Massively parallel simulations of multiphase- and multicomponent flows using lattice Boltzmann methods

Von der
Fakultät Architektur, Bauingenieurwesen und Umweltwissenschaften
der Technischen Universität Carolo-Wilhelmina
zu Braunschweig

zur Erlangung des Grades eines
Doktoringenieurs (Dr.-Ing.)
genehmigte

Dissertation

von
Benjamin Ahrenholz
aus Rostock

Eingereicht am	28. April 2009
Mündliche Prüfung am	17. August 2009
Berichterstatter	Prof. Dr.-Ing. habil. Manfred Krafczyk
	Prof. Dr.-Ing. habil. Rainer Helmig

2009

Eidesstattliche Erklärung

Name: Benjamin Ahrenholz

Geburtsdatum/-ort: 20. Januar 1975, Hansestadt Rostock

Straße: Drostestr. 32

Ort: 30161 Hannover

Hiermit erkläre ich an Eides statt, dass ich die vorgelegte Dissertation mit dem Titel

Massively parallel simulations of multiphase- and multicomponent flows using lattice Boltzmann methods

selbstständig verfasst, nicht schon als Diplom- oder Prüfungsarbeit verwendet und alle in Anspruch genommenen Hilfen in der Dissertation angegeben habe.

Ich versichere an Eides statt, dass ich diesen Promotionsantrag erstmalig einreiche und keine früheren Versuche einer Promotion unternommen habe.

(Ort, Datum, Unterschrift)

Abstract

This thesis reflects the work mainly performed within the research project FIMOTUM focusing on the determination of transport properties and mechanisms in unsaturated media. The efficient simulation of single- and multiphase flows at the pore scale in highly resolved natural porous media is one of the major topics in this work. For this purpose a simulation kernel which is based on the lattice Boltzmann method (LBM) has been developed and extensively validated. The LBM presented utilizes the Multiple Relaxation Time (MRT) model and fluid/wall boundary conditions of second order accuracy. The model has also been extended to solve multiphase, advection/diffusion and thermal flow problems. Due to the application of an optimized collision model and corresponding boundary conditions, the covered parameter space and the stability of the method could be greatly enhanced. Hence, it was possible to perform simulations in complex geometries at a large scale (Degree of Freedom $>$ 2E11) which have been obtained with an unprecedented accuracy.

A second target of this thesis was the design and implementation of a simulation kernel to perform massively parallel computations with high efficiency. In order to obtain accurate simulation results at reasonable computational effort, a novel grid generation procedure has been developed. The robust and flexible method is based on the decoupling of input geometry and the actual computational grid. It is therefore excellently suited for the grid generation based on natural porous media data sets obtained by CT- or X-ray methods. Aspects concerning the increasing difficulties in pre- and post-processing of large data sets are discussed. Furthermore, special issues in high performance computing environments are highlighted and a tool chain to visualize scientific data in photo-realistic representation is described.

Besides the development of model and tool to simulate and analyze multiphase and multicomponent flow problems, various numerical experiments have been performed. First the accuracy of the kernel has been validated by comparing simulation results to reference solutions. Afterwards, single- and multiphase flows in complex geometries have been examined. In this part of the thesis simulation results of drainage and imbibition processes are compared to experiments, finding a good agreement. Furthermore $P_c - S_w$ relationships are determined and effects causing hysteresis are discussed. It is also demonstrated that the simulation kernel is capable to simulate typical engineering problems, e.g. the sound absorption in porous structures or thermal convection in buildings.

Kurzfassung

Die vorliegende Dissertation gibt im Wesentlichen die Arbeiten wieder, die im Rahmen des FIMOTUM Projektes durchgeführt worden sind, welches sich vornehmlich auf die Untersuchung von Transporteigenschaften in ungesättigten porösen Medien fokussierte. Hierfür wurde ein Software-Prototyp auf Basis der Gitter Boltzmann Methode (LBM) entwickelt und ausführlich validiert. Die vorgestellte LB-Methode basiert auf dem Multiple-Relaxation-Time (MRT) Modell und verwendet Fluid/Wand Randbedingungen mit einer Genauigkeit 2. Ordnung. Das beschriebene Modell wurde zudem für die Simulation von Mehrphasen-, Advektion/Diffusions- und Thermalen Problemen erweitert. Durch die Optimierung des Kollisionsmodells und der entsprechenden Randbedingungen konnte der nutzbare Parameterraum deutlich vergrößert werden, so dass Simulationen in komplexen Geometrien mit mehr als 2.0E+11 Freiheitsgraden möglich wurden.

Ein zweites Ziel dieser Arbeit war die Implementierung eines effizienten und hochparallelen Software-Prototypen zur Simulation von fluiddynamischen Problemen. Um möglichst genaue Ergebnisse bei mäßigem Ressourceneinsatz zu erzielen, wurde ein neuartiger Gittergenerierungsprozess entwickelt. Dieses robuste und flexible Verfahren basiert auf der Entkopplung von Eingangsgeometrie und dem eigentlichen Rechengitter. Daher eignet sich dieser Gittergenerator hervorragend für die Erzeugung eines numerischen Gitters aus digitalen Datensätzen natürlicher poröser Medien, wie bspw. Tomographie-Scans. Desweiteren werden, neben allgemeinen Problemen des Hochleistungsrechnens, die zunehmenden Schwierigkeiten bei der Verarbeitung der ständig steigenden Datenmengen im Pre- und Postprocessing diskutiert. Weiterhin wird, unterstützend zur Ergebnisanalyse, eine Prozesskette für die Erzeugung von fotorealistischen Visualisierungen aus Simulationsdaten beschrieben.

Neben der Entwicklung von Modell und Werkzeugen zur Simulation und Analyse von Mehrphasen- und Mehrkomponentenströmungen, wurden im Rahmen dieser Arbeit verschiedene numerische Experimente durchgeführt. Nach der Validierung (dem Vergleich der numerischen Approximation mit einer Referenzlösung) sind eine große Anzahl von Simulationen in natürlichen porösen Strukturen untersucht worden. Hier standen vor allem Drainage- und Imbibitionsprozesse im Vordergrund und der Vergleich der Ergebnisse mit Laborexperimenten, wobei eine gute Übereinstimmung festgestellt werden konnte. Weiterhin wurden Kapillardruck-Sättigungsbeziehungen simuliert und Effekte, die zu Hysterese führen, untersucht und beschrieben. Außerdem wird gezeigt, dass der entwickelte Software-Prototyp sehr gut geeignet ist auch typische ingenieurrelevante Probleme zu simulieren, wie z.B. die Schallabsorption in porösen Strukturen oder thermische Konvektion in Gebäuden.

Acknowledgments

I wish to express my gratitude to all people for their help and support at different stages during the development of this work. Especially thesis supervisor Prof. Dr.-Ing. habil. Manfred Krafczyk primarily for providing academic freedom, for the believe in my abilities, but also for his encouragement, support, and for providing access to state-of-the-art, highest performance, computing infrastructure. Furthermore, I would like to thank Dr.-Ing. habil. Jonas Tölke for his patience and commitment while listening and answering the same questions once in a while and for sharing his knowledge. Also I would like to thank all my colleagues at the Institute of Computational Modeling in Civil Engineering (iRMB) at the Universität Braunschweig for providing an appreciated atmosphere, for valuable discussions and, primarily, for offering their help whenever needed. My very special thanks goes to Prof. Dr.-Ing. habil. Rainer Helmig who kindly offered to be the second examiner (Korreferent) of this thesis.

The results and knowledge gained within the FIMOTUM project (**FI**rst Principle Based **MO**delling of **T**ransport in **U**nsaturated **M**edia) - which was the propelling force behind the evolvement of this thesis - could only be achieved through fruitful cooperations with other research groups working on several scientific fields such as soil physics, computer science as well as data and image analysis. Parallel to the work done with the project partners from the ETH Zürich and the University Stuttgart, important collaborations have also been established with the Helmholtz Centre for Environmental Research and the Paul Scherrer Institute in Switzerland.

Finally and naturally, I want to thank all people - especially Jennifer and Konstantin - which were strongly affected during the writing of this thesis, for their patience, their support and for providing a comfortable and lovely environment. Last but not least I would like to thank my parents for providing the opportunity, since the date of my birth, to achieve a doctoral degree.

This work has been supported by the German Science Foundation (DFG).

Nomenclature

Roman Symbols

A	subset of a sample
Bo	Bond number
$\mathbf{C}$	phase field gradient
c	microscopic reference velocity
C_{index}	chord length distribution
Ca	capillary number
c_d	Drag coefficient
c_s	speed of sound
D	diffusion coefficient
Δt	timestep
Δx	grid spacing
$\mathbf{e}$	unit vector
F	Force
f_i	discrete distribution functions
f_{net}	Fraction of the communication
K	Correction factor
k	wave number
l	Length
M	matrix to transform distributions to moments
M	mobility
m	ratio of the pore space volume to wetted surface
Ma	Mach number
$\boldsymbol{m}^{eq}$	equilibrium moments
P	pressure, performance
r	radius or reference length
Ra	Rayleigh number
Re	Reynolds number
$R_{max,net}$	The restriction in LUPS by the maximum bandwidth of the network
r_{sv}	ratio of pore space volume to wetted surface
S	diagonal matrix consisting of relaxation rates
s_ν	relaxation rate
S	saturation
T	time
u_w	Darcy velocity of the wetting phase
y_1	pipe width

Greek Symbols

α	spatial absorption coefficient
δ	Boundary layer thickness
δp	pressure variation
η	coefficient of shear viscosity

Γ	product of C_x, C_y, and ϕ
κ	permeability
κ^*	dimensionless permeability
λ	wave length
μ	dynamic viscosity
ν	kinematic viscosity
Ω	collision operator
ω	frequency
Φ	phase field ordering parameter
ϕ	porosity
ρ_0	equilibrium density
$\Delta\rho$	density variation
ρ_w, ρ_n	dimensionless density of wetting and non-wetting phase
ϵ_{xx}	moments related to the stress tensor
σ	surface tension
$\sigma_{\alpha\beta}$	Stress tensor
Θ	Contact angle
ζ	bulk viscosity

Superscripts

eq	equilibrium
$e, p_{xx}, p_{ww}, p_{xy}, p_{yz}, p_{xz}$	Moments related to the stress tensor
neq	non equilibrium

Subscripts

ref	reference
i, j, k	spatial coordinate index
Max	maximum value
Med	average value
Min	minimum value
nw	non-wetting Phase
w	wetting Phase
x, y, z	spatial direction

Acronyms

1D	One-Dimensional
2D	Two-Dimensional
3D	Three-Dimensional
API	Application Programming Interface
BCC	Body Centered Cubic
BEM	Boundary Element Method
BGK	Bhatnagar Gross Krook
CFD	Computational Fluid Dynamics
CGI	Computer-generated imagery
CPU	Central Processing Unit
DdQq	d dimensions and q discrete velocities, e.g. D2Q9 or D3Q19
DNS	Direct Numerical Simulation
DoF	Degree of Freedom
FF	inverse fit free form function
FFT	Fast Fourier Transform
FI	First Imbibition

GLB	generalized lattice Boltzmann equation
GPU	Graphics Processing Unit
HTLBE	hybrid thermal lattice Boltzmann equation
JAVA	Programming language
LB	lattice Boltzmann
LBM	lattice Boltzmann method
LES	Large Eddy Simulation
LIBB	Linear Interpolated Bounce Back
SBB	Simple Bounce Back
LUPS	Lattice updates per second
MBW	Maximum memory bandwidth
MBWNet	Maximum network bandwidth
MPI	Message Passing Interface
MPNM	Morphological Pore Network Model
MRT	Multiple-Relaxation-Time
NAS	Network Attached Storage
NB	Bytes per lattice node interacting with memory
NBEx	Bytes per core interacting with memory
NF	Floating point operations per lattice node
nproc	Number of cores/processes
NS	Navier Stokes
NT	Neutron tomography
OpenMP	Shared memory parallel programming API
PD	Primary Drainage
PEAK	Peak performance
POF	Persistence of Vision, an open source ray tracer
PVM	Parallel virtual machine library
RAM	Random access memory
RAT	Ratio of inner to outer nodes
REV	Representative Elementary Volume
SAN	Storage area network
SD	Secondary Drainage
SGS	Sub-grid scale
TLBE	thermal lattice Boltzmann equation
VFC	VIRTUALFLUIDS CLASSIC , a branch of VIRTUALFLUIDS based on uniform grids
vG	inverse fit van Genuchten constrained

List of Figures

List of Tables

Contents

1 Introduction

“Zwei Dinge sind zu unserer Arbeit nötig: Unermüdliche Ausdauer und die Bereitschaft, etwas, in das man viel Zeit und Arbeit gesteckt hat, wieder wegzuwerfen.”
Albert Einstein

1.1 Scope

The work presented in this thesis is divided into three major parts:

- The description of the lattice Boltzmann method originally implemented, as well as the major modifications performed, including extensions and boundary conditions, followed by validation examples and a presentation of a grid decoupling method allowing efficient computations, with second order accuracy in space for complex geometries.
- The presentation of the results obtained from multiphase flow simulations in natural porous media, in order to determine the effective parameters and constitutive relationships. This is combined with comparisons to a morphological pore network model and experiments. Finally, some applications using the advection/diffusion and the thermal extensions are introduced.
- The parallel efficiency, implementation issues as well as pre- and post processing problems are emphasized in the third part. Furthermore, a visualization approach for large datasets based on ray tracing techniques is demonstrated.

While the first part mainly focuses on the physics and numerical modeling aspect of the problem, the second, central part of this thesis, is based on multidisciplinary approaches, like geophysics, Computational Fluid Dynamics, and Computer Science. The last part is located in the fields of high performance computing, scientific visualization, and Computational Engineering.

1.2 Motivation

Predicting the dynamics of fluid flows is a problem of tremendous importance for both industrial and natural processes. Due to the rapid development of supercomputing facilities and advanced algorithms, it becomes more and more feasible to simulate complex fluid flow numerically. During the last decades, various numerical methods have been developed to solve the fundamental Navier-Stokes equations, which describe the motion of viscous incompressible fluids. A new field of research has emerged from these progress in predicting the behavior of fluids: Computational Fluid Dynamics (CFD) [119]. It has become nowadays an important element in (professional) engineering practice, including several branches of engineering (e.g. Aerospace-, Civil-, or Mechanical Engineering).
Fluid flow through porous materials[1] also takes place in many areas of applied science and engineering, e.g. mechanics (acoustics, geomechanics, soil mechanics, rock mechanics), engineering (petroleum-, construction engineering), geosciences (hydrogeology, petroleum geology, geophysics), biology and biophysics, material science, etc. Fluid flow through porous media is a subject of most common interest and has emerged as a separate field of study. Experiments on the microscale (the scale where the porous medium is resolved), whether they are of physical or of numerical nature, commonly focuse on effective parameters and constitutive relationships. Thus, a proper understanding of the processes on the microscale can help improving the prediction of flow behavior on the macro-scale and is extremely important for applied fields like environmental engineering, petroleum industry, irrigation, CO_2 sequestration and many others.

---

[1] A porous material or more commonly a porous medium is a solid permeated by an interconnected network of pores filled with a fluid, which can either be liquid or gaseous. Many natural substances such as rocks, soils, biological tissues (e.g. bones), and man made materials such as cements, foams and ceramics can be considered as porous media.

There are countless applications for multiphase flow like e.g. flow through embankment dams, paper fabrication, composites manufacture, filtering, drying and sintering of iron ore pellets. However, this thesis focuses on soil physics and more specifically, on transport processes in porous media with structures on different scales, which are relevant e.g. for:

- the transport of contaminants such as pesticides or heavy metals in the vadose zone of soils, leading to groundwater contamination,
- assessing the security of nuclear waste disposal sites,
- the description of sites contaminated with non-aqueous phase liquids and the monitoring of appropriate decontamination measures.

The application of CFD methods to porous media fluid flow problems is the main motivation in developing a toolset to simulate single- and multiphase flow, as well as transport processes in arbitrary complex geometries.

In the last two decades, the lattice Boltzmann method has considerably gained momentum in the fields of computational fluid mechanics, kinetic theory, chemical process engineering, and soil mechanics. It has matured to an alternative and efficient numerical scheme for the simulation of fluid flows and transport problems [15, 31, 120, 137] especially in complex geometries. Therefore and during the evolvement of this thesis, the LB method has been chosen as the most promising candidate (naturally besides a variety of other well working methods) to simulate flow problems of arbitrary complexity as demanded by porous media. The ease of parallelization and its high performance potential has been considered as secondary benefits of the method.

Only a few studies [1, 2, 11–13, 88, 101, 150, 164] have reported simulations of multiphase flow in three-dimensional porous medium systems, mainly because of the computational limitations and the complexity of the problem. With the computational resources delivered by supercomputing facilities only, it is possible to perform meaningful simulations for such complex problems.

Computations of constitutive relationships, such as the $P_c - S_w$ relationship in porous media, based on lattice Boltzmann simulations can be found in [7, 95, 113, 150, 157]. The geometry of the porous media was obtained in all cases by tomography methods and the voxel set was derived by image processing methods. The computations carried out in [95, 150] have been performed with small grid sizes and the results are of qualitative nature. [113] calculated the hysteresis of $P_c - S_w$-relationships for a packing of glass beads and obtained good results with respect to experimental data. In a later study [98] the focus moved to the determination of relative permeabilities and by using a more advanced lattice Boltzmann approach more realistic capillary numbers could be reproduced. [157] determined the primary drainage curve (without hysteretic effects) for a porous medium. The medium consisted of sintered glass and the geometry was as well delivered by tomography methods. The measured data have been compared to results of other modeling approaches and good agreement was observed.

It has been pointed out that the understanding of flow processes in porous media is crucial for a wide variety of fields and applications. In order to advance in terms of numerical modeling and simulation this thesis aims on the implementation of a state of the art CFD solver based on the lattice Boltzmann method which is at least - in arbitrary geometries - **second order accurate** for single phase flow in a Stokes type regime. In addition, this implementation should be capable to simulate **multiphase flow** and **advection/diffusion** problems paired with an **excellent parallel efficiency** as well as the possibility to compute simulations with more than $2E11$ DoF.

Prior to the determination of material functions in natural porous media the CFD-code has to be verified. Whereas validation means the assessment of the accuracy of the model - compared to physical reality - by critical experimental tests. Therefore, verification is a crucial part in numerical modeling and particularly useful when compared to analytical predictions made in preferable simple contexts.

Direct computations of flows through porous media are usually based on binarized porous media data mapped to uniform Cartesian grids. The tomographic voxel set is directly used as the computational grid and thus the geometrical representation is usually only first-order accurate due to staircase patterns. In this thesis a more elaborated approach is followed where the input geometry is parameterized allowing to create a numerical grid of arbitrary resolution. This is necessary for computing the subgrid distances which are used to apply higher order fluid/wall boundary conditions to keep the scheme second order accurate in space.

1.3 Outline

The main objective of this thesis is the development of a properly designed tool chain for CFD simulations based on the lattice Boltzmann method, which is capable to run parallel efficient computations of multiphase flow in natural porous media at a large computational scale. Furthermore the determination of material functions in porous materials is a particular focus of this work.

The first part of this thesis (Chapter 2-5) is related to the lattice Boltzmann method from a modeling point of view (Chapter 2) and describes the boundary conditions that have been applied (Chapter 3). The accuracy of the model, as well as the verification of the implementation are discussed in Chapter 4. As a prerequisite for the numerical experiments performed in natural porous media, the grid generation process is introduced in Chapter 5.

The second part of the thesis (Chapter 6-7) examines single and multiphase flow in non-artificial complex geometries (Chapter 6). This part compares drainage- and imbibition experiments (Section 6.8), determines $P_c - S_w$ relationships (Section 6.7), and discusses hysteretic effects (Section 6.6). It is concluded by the description of further numerical experiments utilizing the advection/diffusion and the thermal extensions (Chapter 7).

The third part (Chapter 8-10) highlights special issues of pre- and post processing (Chapter 8) in high performance computing environments and describes a tool chain to visualize scientific data in photo-realistic representation (Section 8.3). Finally, Chapter 10 summarizes the most relevant conclusions and an outlook to future developments and research work is given.

2 The lattice Boltzmann method

> “If I were again beginning my studies, I would follow the advice of Plato and start with mathematics.”
> Galileo Galilei

The lattice Boltzmann method (LBM) is a relatively new simulation technique for complex fluid systems, which has attracted a great deal of interest from researchers in computational physics [15, 31, 51, 120, 137]. Unlike the traditional computational fluid dynamics (CFD), which numerically solves the conservation equations of macroscopic properties (i. e., mass, momentum, and energy), LBM models the fluid consisting of fictitious particles, which perform consecutive propagation and collision processes over a discrete lattice. Due to its particulate nature and local dynamics, LBM is very efficient when dealing with complex boundaries and the incorporation of microscopic interactions.
Historically, LBM originated from the lattice gas automata (LGA), which can be considered as a simplified, fictitious molecular dynamics in which space, time, and particle velocities are all discrete. However, it was discovered that LGA suffers from several native defects including the lack of Galilean invariance (except for $\rho = constant$), the presence of statistical noise and the absence of exponential complexity for three-dimensional lattices. The main motivation for the transition from LGA to LBM was the desire to remove statistical noise by replacing the Boolean particle number in a lattice direction with its ensemble average, the so-called density distribution function. Accompanying this replacement, the discrete collision rules also have to be modified as a continuous function - the collision operator. The first LBM has been proposed by [103] and improved by [72, 73]. However, the connection to the Boltzmann equation (introduced by the Austrian physicist Ludwig Boltzmann in 1872) has been proven afterwards [71, 135]. The Boltzmann equation describes the statistical distribution of one particle in a fluid and the probability to encounter this particle at time t with velocity $\boldsymbol{\xi}$ at location $\mathbf{x}$ [27, 28]:

$$\frac{\partial f}{\partial t} + \boldsymbol{\xi} \cdot \frac{\partial f}{\partial \mathbf{x}} + \mathbf{F} \cdot \frac{\partial f}{\partial \boldsymbol{\xi}} = \Omega(f, f') \tag{2.0.1}$$

In the LBM development, an important simplification is the approximation of the collision operator with the Bhatnagar-Gross-Krook (BGK) relaxation term. This lattice BGK (LBGK) model renders simulations more efficient and allows flexibility of the transport coefficients. On the other hand, it has been shown that the LBM scheme can also be considered as a special discretized form of the continuous Boltzmann equation. Through a Chapman-Enskog expansion [51, 117] or an asymptotic analysis [80], one can recover the governing continuity and Navier-Stokes equations (Equation 2.0.2) from the LBM algorithm [117].

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\nabla)\mathbf{u} = -\frac{1}{\rho}\nabla p + \frac{\mu}{\rho}\Delta \mathbf{u}, \tag{2.0.2a}$$

$$\nabla \mathbf{u} = 0 \tag{2.0.2b}$$

In addition, the pressure field is also directly available from the density distributions and hence there is no additional Poisson equation to be solved as in traditional CFD methods.

A considerable improvement in terms of accuracy and efficiency has been achieved with the introduction of the Multiple-Relaxation-Time model (MRT) [42, 43, 91, 145], where higher order moments are relaxed with different rates than the moments related to the stress tensor. This is only possible by first, transforming the populations of a lattice node into an equivalent moment space. Each moment is associated with one physical quantity like density, momentum, stress tensor and so on. Relevant moments are relaxed with a parameter chosen to match the physical properties of the fluid and the irrelevant moments can be relaxed with a relaxation parameter chosen solely from stability considerations. Another approach to improve the stability and accuracy is the Entropic lattice Boltzmann method which has been developed and improved in [9, 19, 20, 30]. Entropic lattice Boltzmann automata are systems designed to be unconditionally

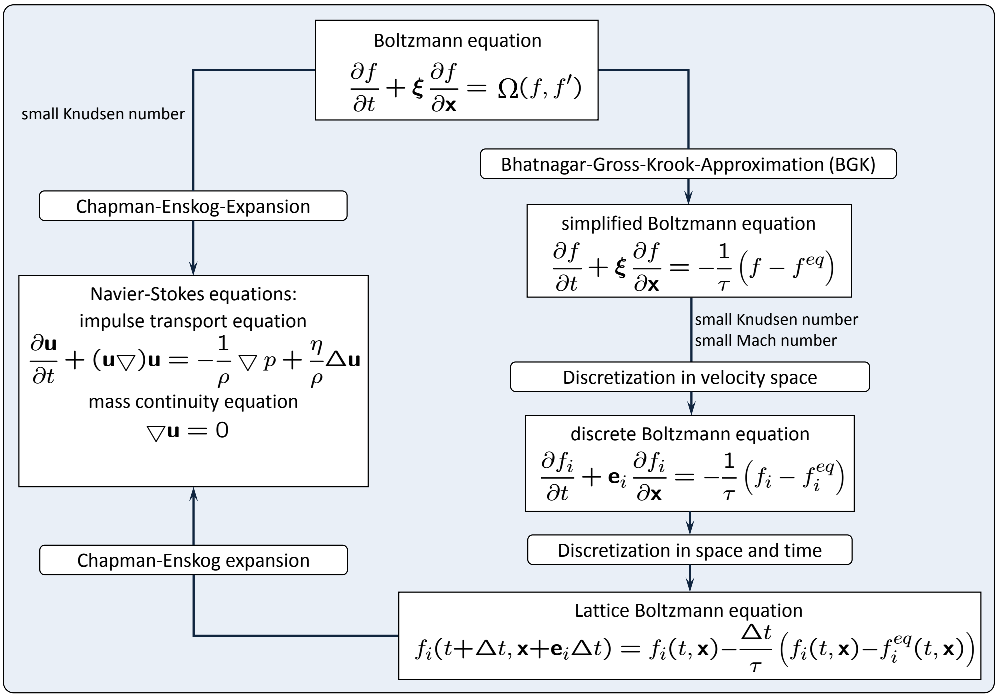


Figure 2.0.1 – From the Boltzmann equation to Navier-Stokes

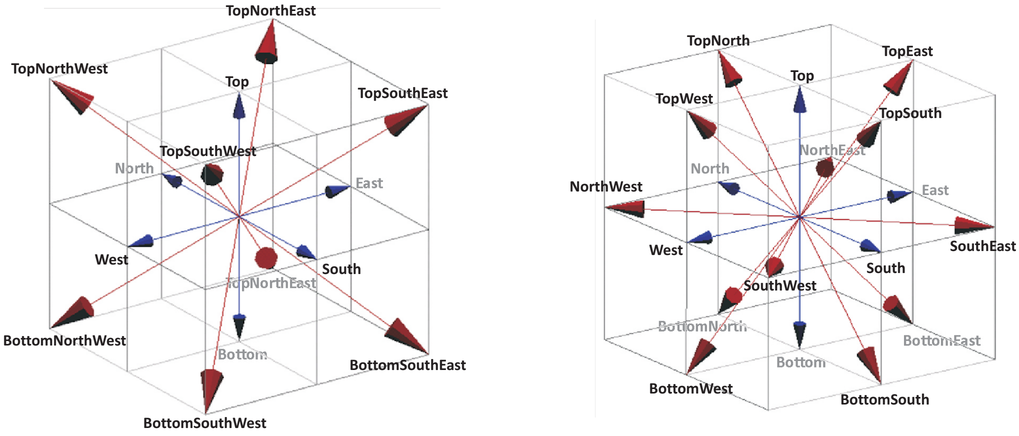
stable despite the fact that the underlying scheme is an explicit time stepping algorithm. The entropic formulation of the lattice Boltzmann method features enhanced numerical stability due to its compliance with the Boltzmann H-theorem. This stability comes at the price of some computational overhead, associated with the need of adjusting the local equilibrium function of the standard LBM in such a way as to secure compliance with the H-theorem.

Simulating multiphase/multicomponent flows has always been a challenge because of the moving and deformable interfaces. More fundamentally, the interfaces between two bulk phases (e. g., oil and water) originate from the specific interactions among different molecules. Therefore, it is difficult to implement such microscopic interactions into the macroscopic Navier-Stokes equation. However, in LBM, the particle kinetics allows a relatively easy and consistent way to incorporate the underlying microscopic interactions by modifying the collision operator. There are three standard lattice Boltzmann models for the simulation of immiscible two-phase systems: The first one has originated from its equivalent LGA model (proposed by [121] and which is usually referred to as the Rothman-Keller (RK) model) and has been developed by [64]. [65] extended the model for binary fluids with different density and viscosity ratios. The second multiphase model is known as the Shan and Chen (SC) model introduced by [130] and is based on mean-field interactions. The third is a free energy approach developed by [138] and usually abbreviated as FE model.

Despite the increasing popularity of LBM in simulating complex fluid systems, one should also be aware of the limitations of this novel approach. At present, high Mach number flows in aerodynamics are still difficult to simulate with LBM, and a consistent thermo-hydrodynamic scheme is still numerically inefficient. For multiphase/multicomponent models, the interfacial thickness is usually large and the density ratio across the interface is limited. Nevertheless, applications and advancements of this method during the past two decades have proven its potential in computational physics, including microfluidics.

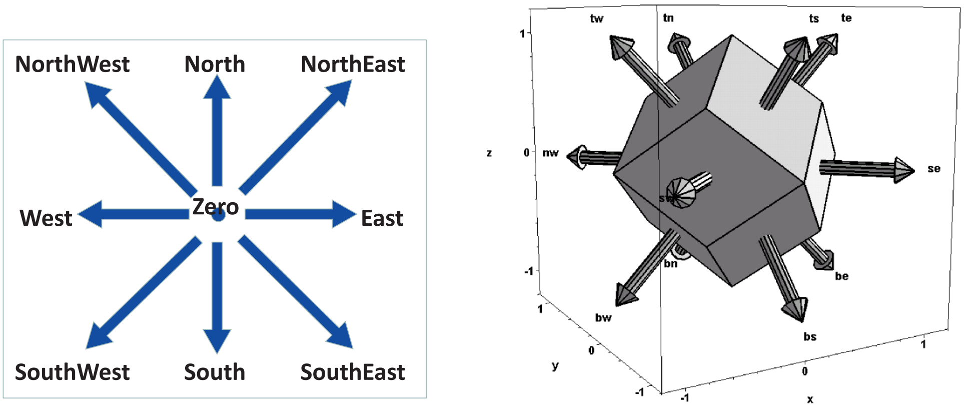
2.1 Multiple-Relaxation-Time model

In this section a short review of the lattice Boltzmann model mainly used throughout this thesis is given. The research software prototype VirtualFluids Classic (VFC) is based on a so called D3Q19 stencil where dxqb [117] is a common labeling for different discretization stencils, where x is the space dimension and b the number of microscopic velocities. Besides D3Q19 models (Figure 2.1.1) one can often find D3Q15 stencils (Figure 2.1.1) in 3D and D2Q9 in 2D (Figure 2.1.2) as well as non local stencils like D3Q27 or D3Q39. D3Q13 uses a reduced set of velocities, however it is very promising due to an excellent ratio between accuracy and speed [40, 147].



(a) D3Q15 stencil (b) D3Q19 stencil

Figure 2.1.1 – D3Q19- and D3Q15 stencils, the most common representatives in 3D



(a) D2Q9 stencil (b) rhombic dodecahedron [40, 147]

Figure 2.1.2 – D2Q9 stencil commonly used for 2D LBM and D3Q13 - the smallest stencil for a space filling grid in 3D

In the following section $\mathbf{x}$ represents a 3D vector in space and $\boldsymbol{f}$ a b-dimensional vector, where b is the number of microscopic velocities. These 19 velocities are given as

$$\mathbf{e}_i, i = 0, \ldots, 18\} =$$

$$\begin{pmatrix} 0 & c & -c & 0 & 0 & 0 & 0 & c & -c & c & -c & c & -c & c & -c & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & c & -c & 0 & 0 & c & -c & -c & c & 0 & 0 & 0 & 0 & c & -c & c & -c \\ 0 & 0 & 0 & 0 & 0 & c & -c & 0 & 0 & 0 & 0 & c & -c & -c & c & c & -c & -c & c \end{pmatrix}$$

where c is a constant microscopic reference velocity related to the speed of sound by $c_s^2 = c^2/3$. The microscopic velocities define a space-filling computational lattice where a node is connected to the neighboring nodes through the vectors $\{\Delta t\,\mathbf{e}_i, i = 0, \ldots, 18\}$. The generalized lattice Boltzmann equation (GLBE) using the Multiple-Relaxation-Time model introduced by [42, 91] is used in VFC in a slightly modified version [146]. These modification are basically a different transformation matrix which is also given in Appendix A. Also, for Stokes flow problems, linearized equilibrium moments are used since viscous forces are dominating inertial forces.
The generalized lattice Boltzmann equation is then given by

$$f_i(t + \Delta t, \mathbf{x} + \mathbf{e}_i \Delta t) = f_i(t, \mathbf{x}) + \Omega_i, \quad i = 0, \ldots, b-1, \tag{2.1.1}$$

where f_i are mass fractions (unit kg m^{-3}) propagating with velocities $\mathbf{e}_i$, Δt is the time step, the grid spacing is $\Delta x = c\Delta t$, and the collision operator of the Multiple-Relaxation-Time model (MRT) is given by

$$\boldsymbol{\Omega} = \mathsf{M}^{-1}\mathsf{S}\left((\mathsf{M}\boldsymbol{f}) - \boldsymbol{m}^{eq}\right). \tag{2.1.2}$$

The matrix M given in appendix A is used to transform the distributions into moment space. The resulting moments $\boldsymbol{m} = \mathsf{M}\boldsymbol{f}$ are labeled as

$$\boldsymbol{m} = (\delta\rho, e, \epsilon, j_x, q_x, j_y, q_y, j_z, q_z, 3p_{xx}, 3\pi_{xx}, p_{ww}, \pi_{ww}, p_{xy}, p_{yz}, p_{xz}, m_x, m_y, m_z),$$

where $\delta\rho$ is a density variation related to the pressure variation δp by

$$\delta p = \frac{c^2}{3}\delta\rho. \tag{2.1.3}$$

and where $(j_x, j_y, j_z) = \rho_0(u_x, u_y, u_z)$ is the momentum and ρ_0 is a constant reference density. The moments $e, p_{xx}, p_{ww}, p_{xy}, p_{yz}, p_{xz}$ are related to the stress tensor:

$$\sigma_{\alpha\beta} = -p\delta_{\alpha\beta} + \rho\nu\left(\frac{\partial u_\alpha}{\partial x_\beta} + \frac{\partial u_\beta}{\partial x_\alpha}\right) \tag{2.1.4}$$

where

$$\sigma_{xx} = -(1 - \frac{s_\nu}{2})(\frac{1}{3}e + p_{xx} - \rho_0 u_x^2) \tag{2.1.5a}$$

$$\sigma_{yy} = -(1 - \frac{s_\nu}{2})(\frac{1}{3}e - \frac{1}{2}p_{xx} + \frac{1}{2}p_{ww} - \rho_0 u_y^2) \tag{2.1.5b}$$

$$\sigma_{zz} = -(1 - \frac{s_\nu}{2})(\frac{1}{3}e - \frac{1}{2}p_{xx} - \frac{1}{2}p_{ww} - \rho_0 u_z^2) \tag{2.1.5c}$$

$$\sigma_{xy} = -(1 - \frac{s_\nu}{2})(p_{xy} - \rho_0 u_x u_y) \tag{2.1.5d}$$

$$\sigma_{yz} = -(1 - \frac{s_\nu}{2})(p_{yz} - \rho_0 u_y u_z) \tag{2.1.5e}$$

$$\sigma_{xz} = -(1 - \frac{s_\nu}{2})(p_{xz} - \rho_0 u_x u_z). \tag{2.1.5f}$$

Here s_ν is a collision rate which is described below. The other moments of higher order have no physical meaning for the Navier-Stokes equations. The vector $\boldsymbol{m}^{eq}$ is composed of the equilibrium moments given by

$$m_0^{eq} = \delta\rho \tag{2.1.6a}$$

$$m_1^{eq} = e^{eq} = \rho_0\,(u_x^2 + u_y^2 + u_z^2) \tag{2.1.6b}$$

$$m_3^{eq} = \rho_0\,u_x \tag{2.1.6c}$$

$$m_5^{eq} = \rho_0\,u_y \tag{2.1.6d}$$

$$m_7^{eq} = \rho_0\,u_z \tag{2.1.6e}$$

$$m_9^{eq} = 3p_{xx}^{eq} = \rho_0\,(2\,u_x^2 - u_y^2 - u_z^2) \tag{2.1.6f}$$

$$m_{11}^{eq} = p_{zz}^{eq} = \rho_0\,(u_y^2 - u_z^2) \tag{2.1.6g}$$

$$m_{13}^{eq} = p_{xy}^{eq} = \rho_0\,u_x\,u_y \tag{2.1.6h}$$

$$m_{14}^{eq} = p_{yz}^{eq} = \rho_0\,u_y\,u_z \tag{2.1.6i}$$

$$m_{15}^{eq} = p_{xz}^{eq} = \rho_0\,u_x\,u_z \tag{2.1.6j}$$

$$m_2^{eq} = m_4^{eq} = m_6^{eq} = m_8^{eq} = m_{16}^{eq} = m_{17}^{eq} = m_{18}^{eq} = 0 \tag{2.1.6k}$$

The matrix S is a diagonal collision matrix composed of relaxation rates $\{s_{i,i}, \ldots, b-1\}$, also called the eigenvalues of the collision matrix $\mathsf{M}^{-1}\,\mathsf{S}\,\mathsf{M}$. The rates different from zero are

$$\begin{aligned}
s_{1,1} &= -s_e\\
s_{2,2} &= -s_\epsilon\\
s_{4,4} &= s_{6,6} = s_{8,8} = -s_q\\
s_{10,10} &= s_{12,12} = -s_\pi\\
s_{9,9} &= s_{11,11} = s_{13,13} = s_{14,14} = s_{15,15} = -s_\nu\\
s_{16,16} &= s_{17,17} = s_{18,18} = -s_m.
\end{aligned}$$

The relaxation rate s_ν is related to the kinematic viscosity ν by

$$s_\nu = \frac{1}{3\frac{\nu}{c^2\Delta t} + \frac{1}{2}}. \tag{2.1.7}$$

The remaining relaxation rates $s_e, s_\epsilon, s_q, s_\pi$ and s_m can be freely chosen in the range of $[0, 2]$ and may be tuned to improve accuracy as well as stability [91] of the model. The optimum values depend on the specific system under consideration (geometry, initial, and boundary conditions) and can therefore not be computed in advance for general cases. For Stokes flow a good choice are the 'magic' parameters relaxing the even and the odd moments differently [58]:

$$s_e = s_\xi = s_\pi = s_\nu, \quad s_q = s_m = 8\frac{(2-s_\nu)}{(8-s_\nu)} \tag{2.1.8}$$

These parameters eliminate the dependence on permeability[1] on s_ν - for simulations using the simple bounce-back scheme as boundary conditions and reduce them substantially in the case

---

[1] Unfortunately, in typical lattice Boltzmann models the permeability of a structure depends on the chosen viscosity if the Reynolds number remains the same. This is an error introduced by the fluid/wall boundary conditions. As a strongly simplified explanation one can state that the distance of the wall between a fluid and a solid node is modified by the chosen viscosity. However, this error can be eliminated completely for special cases or at least reduced substantially by a proper choice of relaxation parameters [58, 61].

of interpolation based bounce back schemes. Furthermore, some reasonable values for these parameters are also given in [43] and a common set of values frequently used for the MRT-model is $s_a = s_b = s_c = s_d = s_e = \max\{s_\omega, -1.0\}$. If different values where chosen for particular numerical examples a more detailed description is given in the corresponding section.
In summary it may be noted that if one uses either a Chapman-Enskog expansion [51] or an asymptotic expansion using the diffusive scaling [80], it can be shown that the lattice Boltzmann method is a scheme of first order in time and second order in space for the incompressible Navier-Stokes equations.

2.2 The multiphase extension

In order to simulate immiscible two phase flow one has to extend the single phase approach described in the preceding section. Therefore a wetting and non-wetting dimensionless density field ρ_w^* and ρ_n^* is introduced and an order parameter ϕ is defined

$$\phi = \frac{\rho_w^* - \rho_n^*}{\rho_w^* + \rho_n^*}, \tag{2.2.1}$$

which indicates the fluid phase: a value of $\phi = 1$ represents the wetting phase and $\phi = -1$ the non-wetting phase. The value of ϕ is constant in the bulk of each phase and varies only in the diffusive fluid-fluid interface. The gradient $\mathbf{C}$ of ϕ is then computed by

$$\mathbf{C}(t, \mathbf{x}) = \frac{3}{c^2\,\Delta t} \sum_i w_i\, \mathbf{e}_i\, \phi(t, \mathbf{x} + \mathbf{e}_i \Delta t). \tag{2.2.2}$$

The normalized gradient

$$n_\alpha = \frac{C_\alpha}{|\mathbf{C}|}, \tag{2.2.3}$$

defines the orientation of the fluid-fluid interface and is also used to generate surface tension (see equations 2.2.7b-2.1.6k). The advection of the density fields $\phi = \{\rho_w^*, \rho_n^*\}$ is done with the following LB equation:

$$g_i(t + \Delta t, \mathbf{x} + \mathbf{e}_i \Delta t) = g_i^{eq}(\psi(t, \mathbf{x}), \mathbf{u}(t, \mathbf{x})) \tag{2.2.4}$$

where g_i are dimensionless density distributions. The equilibrium distribution function g_i^{eq} is given by

$$g_i^{eq}(\psi, \mathbf{u}) = w_i\, \psi\, (1 + \frac{3}{c^2} \mathbf{e}_i \cdot \mathbf{u}). \tag{2.2.5}$$

and the velocity $\mathbf{u}$ is computed by the flow solver described in section 2.1. The weights w_i are

$$w_i = \begin{cases} \frac{1}{3} & \text{for } i = 0 \\ \frac{1}{18} & \text{for } i = 1,2,3,4,5,6 \\ \frac{1}{36} & \text{for } i = 7,8,9,10,11,12,13,14,15,16,17,18 \end{cases} . \tag{2.2.6}$$

The scheme (2.2.4) in combination with (2.2.5) results in an advection/diffusion equation [83]. The diffusion coefficient is $\alpha = \frac{1}{6}c^2\Delta t$ which is undesirable for immiscible fluids and a recoloring algorithm is applied to cancel out the diffusion effect.

The recoloring algorithm to separate phases:
A recoloring step is introduced to eliminate the diffusion effects and to achieve a phase separation. The recoloring step redistributes the distributions g_i *of phase* ρ_w^* *and* ρ_n^* *so that the inner product of the gradient* $\mathbf{C}$ *and the momentum of phase* ρ_w^* *is maximized. The constraints are the conservation of the mass of each phase and the conservation of the momentum of the sum of both phases. The recoloring algorithm proposed in [148, 150] is used.*

One of the advantages of this multiphase approach is that for the scheme given by (2.2.4) only ϕ as ordering parameter of the phase field and the advection velocity $\mathbf{u}$ as input value is needed. Therefore it is not necessary to store nineteen distribution functions for each field, instead it is sufficient to store only the two variables ρ_w^* and ρ_n^*.

While the flow field transport is furthermore based on the scheme from 2.1 the equilibrium moments are extended by terms responsible for the generation of surface tension and are also optimized for Stokes flow problems. The vector $\boldsymbol{m}^{eq}$ is composed of these moments and given by

$$m_0^{eq} = \delta\rho \tag{2.2.7a}$$

$$m_1^{eq} = e^{eq} = -\sigma\,|\mathbf{C}| \tag{2.2.7b}$$

$$m_3^{eq} = \rho_0\,u_x \tag{2.2.7c}$$

$$m_5^{eq} = \rho_0\,u_y \tag{2.2.7d}$$

$$m_7^{eq} = \rho_0\,u_z \tag{2.2.7e}$$

$$m_9^{eq} = 3p_{xx}^{eq} = \frac{1}{2}\sigma|\mathbf{C}|\,(2\,n_x^2 - n_y^2 - n_z^2) \tag{2.2.7f}$$

$$m_{11}^{eq} = p_{zz}^{eq} = \frac{1}{2}\sigma|\mathbf{C}|\,(n_y^2 - n_z^2) \tag{2.2.7g}$$

$$m_{13}^{eq} = p_{xy}^{eq} = \frac{1}{2}\sigma|\mathbf{C}|\,(n_x n_y) \tag{2.2.7h}$$

$$m_{14}^{eq} = p_{yz}^{eq} = \frac{1}{2}\sigma|\mathbf{C}|\,(n_y n_z) \tag{2.2.7i}$$

$$m_{15}^{eq} = p_{xz}^{eq} = \frac{1}{2}\sigma|\mathbf{C}|\,(n_x n_z) \tag{2.2.7j}$$

$$m_2^{eq} = m_4^{eq} = m_6^{eq} = m_8^{eq} = m_{16}^{eq} = m_{17}^{eq} = m_{18}^{eq} = 0 \tag{2.2.7k}$$

Here $\mathbf{C}$ is the gradient of the phase field and $\mathbf{n}$ the normalized gradient indicating the orientation of the fluid-fluid interface. The definitions are also given in equation (2.2.2) and (2.2.3). The moments $\{m_k, k = 0, 3, 5, 7\}$ are conserved during the collision, leading to mass and momentum conservation of the algorithm. The matrix S is the same diagonal collision matrix composed of relaxation rates as already introduced during the presentation of the MRT model in Section 2.1.

The moments $e, p_{xx}, p_{ww}, p_{xy}, p_{yz}, p_{xz}$ are related to the strain rate tensor $\epsilon_{\alpha\beta} = \partial_\beta\ u_\alpha + \partial_\alpha\ u_\beta$ by

$$\epsilon_{xx} = -(1 - \frac{s_\nu}{2})(\frac{e}{3} + p_{xx} + \sigma\ |\mathbf{C}|\ (n_y^2 + n_z^2)) \tag{2.2.8a}$$

$$\epsilon_{yy} = -(1 - \frac{s_\nu}{2})(\frac{e}{3} - \frac{1}{2}p_{xx} + \frac{1}{2}p_{ww} + \sigma\ |\mathbf{C}|\ (n_x^2 + n_z^2)) \tag{2.2.8b}$$

$$\epsilon_{zz} = -(1 - \frac{s_\nu}{2})(\frac{e}{3} - \frac{1}{2}p_{xx} - \frac{1}{2}p_{ww} + \sigma\ |\mathbf{C}|\ (n_x^2 + n_y^2)) \tag{2.2.8c}$$

$$\epsilon_{xy} = -(1 - \frac{s_\nu}{2})(p_{xy} - \sigma\ |\mathbf{C}|\ (n_x n_y)) \tag{2.2.8d}$$

$$\epsilon_{yz} = -(1 - \frac{s_\nu}{2})(p_{yz} - \sigma\ |\mathbf{C}|\ (n_y n_z)) \tag{2.2.8e}$$

$$\epsilon_{xz} = -(1 - \frac{s_\nu}{2})(p_{xz} - \sigma\ |\mathbf{C}|\ (n_x n_z)). \tag{2.2.8f}$$

A detailed mathematical analysis shows that this numerical model yields the Navier-Stokes equations (Equation 2.0.2) for two immiscible phases with surface tension [83]. The scheme is formally first order in time and second order in space. Note that the extensions to the original method [64], especially the linear advection scheme, the moment method and the altered terms (2.2.7b-2.1.6k) for imposing surface tension, substantially improve the numerical efficiency. A method for adjusting the contact angle between the wetting and non-wetting phase is proposed in [148].

2.3 Advection/diffusion extension

Early lattice Boltzmann models for solving the advection-diffusion equation have been described by [47] and [163]. Later, [153] developed a scheme which also works on irregular Bravais lattices. Recently, [57] described a family of lattice Boltzmann schemes for the advection/diffusion equation also allowing for anisotropic diffusion and providing many parameters allowing a fine tuning of the numerical properties of the method. In [136] a coupling algorithm is introduced which allows the extension of the advection/diffusion model to multiply locally refined grids. In the following a basic description of an advection/diffusion approach for the lattice Boltzmann model is given.

If one wants to solve the advection/diffusion equation

$$\frac{\partial s}{\partial t} + u_\alpha \frac{\partial s}{\partial x_\alpha} = D \frac{\partial^2 s}{\partial x_\alpha \partial x_\alpha} \tag{2.3.1}$$

with an externally given velocity field $\mathbf{u}(\mathbf{x}, t)$ and diffusion coefficient $D(\mathbf{x}, t)$. The basis of the algorithm is one particular member of the family of advection/diffusion LB algorithms described in [57], where it is labeled as the ‘optimal convection solution ’. Here the D3Q19 set [117] of microscopic velocities $\boldsymbol{\xi}_q$ (Figure 2.1.1) is used.

$$\boldsymbol{\xi}_q = c \begin{cases} (0,0) & : \quad q = 0 \\ (\cos\left[(q-1)\pi/2\right], \sin\left[(q-1)\pi/2\right]) & : \quad q = 1, 2, ..., 19 \end{cases} \tag{2.3.2}$$

The collision operator reduces to a BGK-type [16] relaxation. Thus, the basic lattice Boltzmann algorithm is given as

$$f_q\left(\mathbf{r} + \Delta t \boldsymbol{\xi}_q, t + \Delta t\right) = f_q\left(\mathbf{r}, t\right) + \lambda \left[f_q\left(\mathbf{r}, t\right) - f_q^{\mathrm{eq}}\left(\mathbf{r}, t\right)\right], \tag{2.3.3}$$

which can be interpreted as a local collision step and propagation of the post-collision distributions to the neighboring nodes. Following [57], the collision eigenvalue λ is chosen as

$$\lambda = -3 + \sqrt{3} \tag{2.3.4}$$

to provide the optimal convection solution. The equilibrium distributions are given by

$$f_q^{\mathrm{eq}} = s \left(t_q + \frac{1}{3} a^{(e)} t_q^{(e)} + E_q^u + \frac{1}{2} \frac{u_\alpha \xi_{q\alpha}}{c^2} \right) \tag{2.3.5}$$

with

$$s = \sum_{q=0}^{q=18} f_q = \sum_{q=0}^{q=18} f_q^{\mathrm{eq}}. \tag{2.3.6}$$

$a^{(e)}$ is connected to the diffusion coefficient D via

$$a^{(e)} = \frac{6\sqrt{3}D}{c^2 \Delta t} - 1. \tag{2.3.7}$$

The weights t_q are given by

$$t_0 = \frac{1}{3}, t_q = \frac{1}{6} \text{ for } q \neq 0. \tag{2.3.8}$$

$t_q^{(e)}$ is a basis vector which can be obtained from a second order polynomial in $\boldsymbol{\xi_q}$:

$$t_0^{(e)} = -2, t_q^{(e)} = \frac{1}{2} \text{ for } q \neq 0. \tag{2.3.9}$$

E_q^u is used to remove a velocity dependent second order correction to the diffusion tensor:

$$E_0^u = -\frac{u_\alpha u_\alpha}{c^2}, E_q^u = \frac{1}{2} \frac{(u_\alpha \xi_{q\alpha})^2}{c^4} \text{ for } q \neq 0 \tag{2.3.10}$$

The verification of this approach, as well as for uniform and for non-uniform grids, has been demonstrated [136]. It is shown that this approach allows an efficient solution of advection/diffusion problems on uniform grids and also on locally refined Cartesian meshes.

2.4 Hybrid Thermal LBM

To simulate thermal flows in engineering applications the basic LBM has to be extended due to the absence of an inherently consistent thermal model, respectively an empty moment which is responsible for energy conservation.

During the last decade different approaches for the simulation of thermal driven flows using the LB method have been developed [8, 46, 93, 116, 129, 152]. Energy conserving thermal LB equation models (TLBE) use a larger set of discrete velocities than the standard method [117] to include a thermal variable, such as temperature. The internal energy is defined through the second-order moment of the distribution function, and the collision operator is chosen to satisfy local energy conservation. However, these thermal flow simulations utilizing the thermal lattice Boltzmann equation (TLBE) are hampered by numerical instabilities caused by an algebraic coupling among different modes of the linearized collision operator, independently of the number of discrete velocities used in the model [93]. To avoid the limitations of TLBE a hybrid scheme was developed by [93] by coupling the energy mode of the athermal LB model to the temperature

field. This method has been extended for turbulent flows by [155] and is usually referred to as hybrid thermal lattice Boltzmann equation (HTLBE).
In this work the temperature equation is discretized by the following finite difference (FD) scheme:

$$\frac{T_{i,j,k}(t+\Delta t^{FD}) - T_{i,j,k}(t)}{\Delta t^{FD}} = -\vec{j}_{i,j,k}(t)\nabla^{(h)}_{i,j,k} T_{i,j,k}(t) + \alpha \triangle^{(h)}_{i,j,k} T_{i,j,k}(t) \tag{2.4.1}$$

where α is the thermal diffusivity. For computing the difference operators $\nabla^{(h)}_{i,j,k}$ and $\triangle^{(h)}_{i,j,k}$ a 6 point stencil is used. The coupling of both schemes is explicit, meaning that the velocity field obtained by the MRT scheme is inserted into the energy equation while the solution of the latter is used to compute the buoyant force $F_z(\vec{x},t)$ in the sense of a Boussinesq approximation. For a given Rayleigh number

$$Ra = \frac{Pr g_z \beta \Delta T L^3}{\nu^2} \tag{2.4.2}$$

and Prandtl number $Pr = \nu/\alpha$ and by setting $\beta = 1/T_0$, the parameters viscosity ν and diffusivity α are obtained and the relaxation coefficients can be determined with the formulae given in [93], while ν has to fulfill the stability constraints of the MRT scheme. L is a characteristic length scale of the dimensionless system, i.e. given in lattice units. The coupling of the temperature field to the energy mode of the LB model is done by inserting the temperature into the equilibrium moments [145]:

$$m_1^{eq} = ((3T-1) + (u_x^2 + u_y^2 + u_z^2))\rho_0 \tag{2.4.3}$$
$$m_2^{eq} = (1 - 1.8T)\rho_0 \tag{2.4.4}$$

where $T = T(t,i,j,k)$ is a dimensionless temperature varying in space and time.

In order to simulate more realistic engineering applications, such as convective heat transport in buildings, simulations with Reynolds numbers of more than 10^6 have to be performed. At this scale DNS simulations become challenging and therefore it is necessary to extend the standard HTLBE by a turbulence model. Large-eddy (LES) approaches are regarded as a promising compromise between explicit modeling of all scales of the turbulent spectrum and direct numerical simulation (DNS). In LES the large scale motions of the flow are calculated, while the effect of the smaller universal scales (the so called sub-grid scales) are modeled using a sub-grid scale (SGS) model. The most commonly used SGS model is the Smagorinsky model. It compensates for the unresolved turbulent scales through the addition of an “eddy viscosity” into the governing equations.
In the context of lattice Boltzmann, the LES approach has first been used by [75] in 2D and in [89] in 3D. As an inherent property of the LBE scheme, components of the momentum flux tensor, here expressed in terms of moments,

$$\Pi_{\alpha\beta} = \sum_i e_{i\alpha} e_{i\beta} f_i \tag{2.4.5}$$

are given as local quantities and do not have to be computed from derivatives of hydrodynamic quantities. Therefore, the local strain tensor is obtained by the relation

$$\tilde{\epsilon}_{\alpha\beta} = \frac{s_{xx}}{2\rho C_s^2}(C_s^2 \rho \delta_{\alpha\beta} + \rho u_\alpha u_\beta - \Pi_{\alpha\beta}) \tag{2.4.6}$$

as previously shown by [89]. Consequently, the molecular and turbulent viscosities can be added to form a total viscosity $\nu_{total} = \nu_0 + \nu_T$ which substitutes the material property by a space and time-dependent quantity. Having computed a local value for ν_T, the relaxation parameter s'_{xx} for the second order moments related to stress $3p_{xx}, p_{ww}, p_{xy}, p_{yz}$ and p_{zx} can be determined by

$$s'_{xx} = \frac{1}{3(\nu_0 + \nu_T) + \frac{1}{2}} \quad , \tag{2.4.7}$$

where ν_T is either related to the same time step or the last time step before propagation of the explicit scheme. Due to consistency reasons a subgrid model is also used for heat flux as proposed in [155].

3 Boundary conditions

The governing equations of a flow problem, it is necessary to specify conditions that model the behavior of the fluid and flow properties at boundaries of various types. The boundary conditions that apply specifically to the velocity field are referred to as the no-slip boundary conditions (section 3.1). In nearly all flows encountered in engineering, observation shows that a fluid does not move relative to a solid surface in the tangential direction. However, sometimes it may be useful to mimic an interface condition with a semi slip or even a free slip condition to model e.g. unbound fluids which are described in section 3.2. A computational flow domain can either be completely sealed off, e.g. with non permeable walls and their corresponding boundary conditions or, in most cases, they are influenced by the environment. Two types of boundaries are introduced: the periodic space as used for an unbounded domain (section 3.4) and a Dirichlet type boundary condition used for in- and outflow into the computational domain (section 3.3). Besides well defined boundary conditions, certain engineering problems require the presence of a body force, such as gravity. This body force is introduced in section 3.5.

3.1 No-Slip Boundary Conditions

For friction dominated flow, observation shows that a fluid does not move relative to a solid surface in the tangential direction. Rather, the fluid sticks to the surface, a phenomenon referred to as no-slip. Conceptually, one can think of the outermost molecules of the fluid stick to the surfaces past which it flows. From this one concludes that the tangential component of velocity u_T is equal to the tangential component of boundary velocity U_T. This boundary condition

$$u_T = U_T \tag{3.1.1}$$

is called the no-slip condition in fluid dynamics. In lattice Boltzmann models the macroscopic flow quantities can only be set implicitly via incoming particle distribution functions on the boundary nodes. A well known and simple way to introduce no-slip walls is the so-called bounce back scheme which allows spatial second order accuracy if the boundary is aligned with one of the lattice vectors $\vec{e}_i$ and first order otherwise [60]. The focus in this work aims on flows in porous media which are arbitrary shaped objects and their geometric representation is either first order accurate if present as black and white scans or even second order accurate if present as grey scale matrices. In order to conserve the accuracy of the geometric representation, we use the modified bounce back scheme developed in [21, 92] for velocity boundary conditions, which are second order accurate for arbitrarily shaped boundaries (Figure 3.1.1).

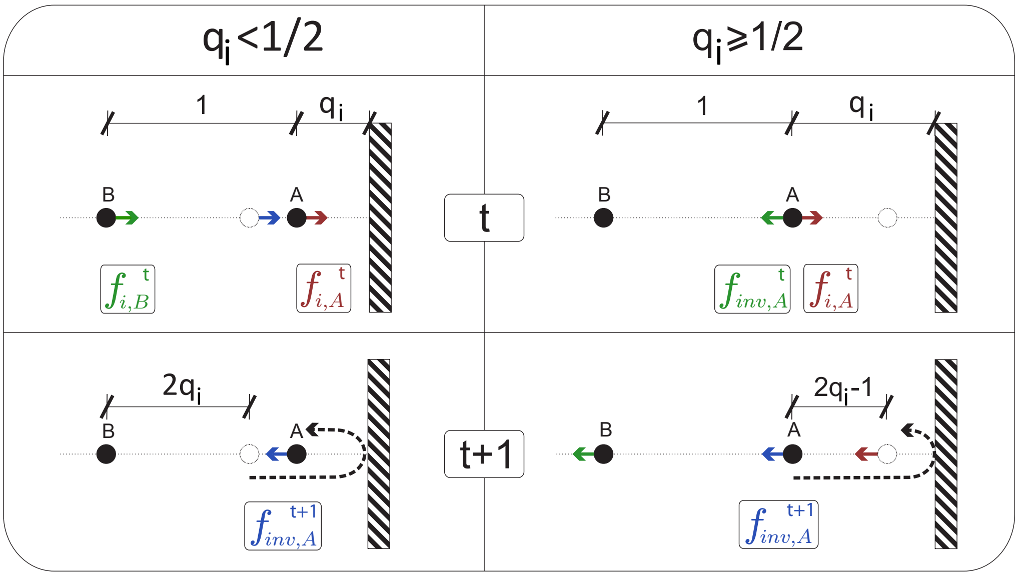


Figure 3.1.1 – Interpolation scheme based on subgrid distances for bounce back with second order accuracy.

Here we identify two cases:

(i) the wall has a distance less than 0.5 $\mathbf{e}_i \Delta t$ from the node:

$$0.0 < q_i < 0,5: \; f_{inv,A}^{t+1} = (1 - 2q_i)\; f_{i,B}^t + 2q_i \; f_{i,A}^t - 2\rho w_i \; \frac{\mathbf{e}_i \mathbf{u}_w}{c_s^2} \tag{3.1.2}$$

(ii) the wall has a distance between 0.5 and 1.0 $\mathbf{e}_i \Delta t$ from the node:

$$0,5 \leq q_i \leq 1,0: \; f_{inv,A}^{t+1} = \frac{2q_i - 1}{2q_i}\; f_{inv,A}^t + \frac{1}{2q_i}\; f_{i,A}^t - \rho w_i \; \frac{\mathbf{e}_i \mathbf{u}_w}{q_i \; c_s^2} \tag{3.1.3}$$

The distributions at time $t/t+1$ are post/pre-collision values, $q\mathbf{e}_i \Delta t$ is the distance to the wall and $\mathbf{u}_w$ the velocity at the wall. Therefore we obtain second order accurate results in space even for curved geometries [54]. For a detailed discussion of LBE boundary conditions we refer to [60]. In contrast to the simple bounce-back scheme, the use of this interpolation based no-slip boundary conditions result in a notable mass loss across the no-slip lines. To circumvent this problem the mass difference is transferred to the rest particle distribution. This results in a no slip scheme which is conservative in mass (and thus pressure) while introducing a higher-order disturbance of the stress tensor, however, this does not change the results significantly. The results obtained with the first-order bounce-back were inferior to the second-order scheme which highlights the importance of a proper geometric resolution of the flow domain.

3.2 Free Slip Boundary Conditions

One important exception to the no-slip condition occurs if the fluid is modeled as inviscid or if the flow boundary is modeled as a frictionless wall. In either case the flow appears to be frictionless in the vicinity of the interface, and there are no shear forces acting at the solid/fluid interface.

The fluid is assumed to have the ability to slip freely in any tangential direction along the solid surface. While the bounce back rule models a no-slip non permeable wall boundary condition, a free slip boundary on the other hand models non permeable frictionless walls. Despite this behavior is very rare in natural systems (supra viscous fluids or near frictionless surfaces), these type of fluid-wall interface is desirable in many applications of computational fluid dynamics. Some of them are:

- advection/diffusion flux boundary conditions,
- flows at high Reynolds numbers
- accelerated flows,
- non viscous or Euler flows,
- fluid/fluid interfaces.

Another case of application is the transition of a multiphase system into a single-phase context. Here, a fully developed multiphase system (e.g. a saturated porous medium, where air is displacing water) can be modeled by a good approximation into a single phase system where the non-wetting phase (air) is transformed into a solid phase and the resulting interface is endowed with a free slip boundary condition. That facilitates the computation of e.g. relative permeabilities, where the non-wetting phase is eventually immobile.

The variety of free slip boundaries for lattice Boltzmann models in the literature is limited, however, recently some progress has been made [6, 77, 139]. First approaches for modeling slip boundary conditions were either first order accurate or they did work only with plane walls aligned to the computational grid [59]. A new approach using the normal of the fluid/solid interface to project the tangential velocity component to the wall, mimicking a frictionless interface, has been described by B. Ahrenholz, J. Tölke, and M. Krafczyk [6] at the DSFD conference in 2006. Independently in his PhD-thesis [77] based on the work of [139] described a comparable method using the normal information to compute the velocity at the wall boundary. The idea behind this approach is simple: the velocity component corresponding to the anticipated tangential velocity at the wall is extrapolated utilizing the normal of the fluid/wall interface (Figure 3.2.1). The obligatory precondition is the presence of accurate normal data which has been generated by the grid generation process described in section 5.4. [6] demonstrated that the chosen approach works with arbitrary shaped geometries and is second order accurate if the implementation uses interpolation techniques such as described in section 3.1. Validation examples can be found in section 4.1 and an application example in section 6.3.

3.3 In- and outflow

To impose a flow-field into a defined system where the initial velocities are zero and the pressure is uniformly distributed, either Dirichlet- or Neumann type boundary conditions or a combination of both are suitable to uniquely define the flow problem. In the meantime it is well known how to model and implement a Dirichlet type [76, 133, 167], whereas more effort is required to apply a Neumann boundary condition, because one has to compute the gradients of a pressure or velocity field and therefore the implementation will be non local. Therefore the boundary types for imposing motion are typically restricted to either body force 3.5 or pressure- as well as velocity boundary conditions of Dirichlet type. In most cases the pressure likewise the velocity

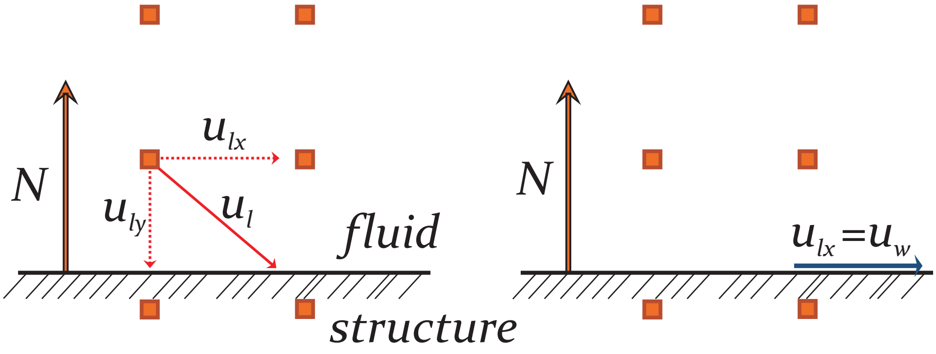


Figure 3.2.1 – Free slip boundary

is directly set at the boundary to impose a gradient driving the flow over the whole domain. It is noted that is it not useful to set pressure and velocity at the same time, except the solution to the problem is known and one can input the proper values.
In summary, for all types of problems computed in the focus of this thesis, the issue of in- and outflow conditions is basically reduced to either set velocity or pressure for in- respectively outflow boundary planes.

3.4 Periodic space

In fluid mechanics periodic boundary conditions are a set of boundary conditions used to simulate an unbounded system in at least one geometric dimension. They are particularly useful for simulating a part of a bulk system, mimicking two dimensions, while leaving the third dimension open or for simulating an infinitely long channel, duct or tube. These types of boundary conditions are implemented easily; at least into a simulation kernel already suited with parallel capabilities, because the outflow can be targeted instead to a neighboring domain, to the inflow of the corresponding domain. While using lattice Boltzmann methods one has to assure that past the propagation step the missing in- and outgoing distributions at the boundary are exchanged as exemplarily shown for the x1 direction in a 2D (D2Q9) domain:

$$f_{(NE,E,SE)}(t,(0,x2)^T) = \tilde{f}_{(NE,E,SE)}(t,(L_{x1},x2)^T); \tag{3.4.1}$$

$$f_{(NW,W,SW)}(t,(L_{x1},x2)^T) = \tilde{f}_{(NW,W,SW)}(t,(0,x2)^T); \tag{3.4.2}$$

This has been implemented - using an MPI approach - for the simulation kernel mainly used throughout this thesis (a more detailed insight is given in chapter 9). To simplify data exchange so called ghost or halo nodes have been used (section 9.1).
While applying periodic boundary conditions one has to be aware that disturbances also travel freely through the whole domain. Also it is noted, that the pressure distribution may not be accurate, because the pressure distribution cannot fully equilibrate: e.g. if bubbles are simulated inside an unbounded fluid and the applied surface tension causes the development of capillary pressure and therefore a decreasing pressure in the remaining domain. This negative pressure respectively the mass loss can not be compensated if periodic boundary conditions are imposed.

3.5 Volume Forces

So far external forces have been neglected. However, one of the most important ones which is always present is gravity . This force is impellent for various types of flows e.g. drainage water flows, free surface flows or even for flows where buoyancy effects are taking place (rising bubbles). In lattice Boltzmann models one applies a body force to the fluid by adding a forcing term either to the equilibrium distribution function:

$$\delta f_i = 3\,\omega_i\,\rho \sum_i \mathbf{e}_i \mathbf{F} \Delta t \tag{3.5.1}$$

or directly into moment space to the corresponding velocity moments, while using a Multiple-Relaxation-Time model:

$$\delta \boldsymbol{j}_i = \rho \mathbf{F_i} \Delta t \tag{3.5.2}$$

Several approaches of modeling volume forces have been made as well as some work regarding the analysis of how to implement them correctly [23, 56, 60, 66, 90]. However, an extensive overview and comparison as well as an detailed analysis has recently been published by [61] showing that the result of basically all approaches for applying volume forces are comparable.

3.6 Initial conditions

If information about velocity and pressure are not known, initial conditions can be defined by starting from an equilibrium distribution function. The flow density everywhere on the grid is equal to a constant:

$$\rho(\mathbf{x}, t) = m \sum_{i=0}^{n} f_i^{eq}(\mathbf{x}, t) \tag{3.6.1}$$

as well as the speed is equal to zero at each node in the system before the first collision and propagation step applies. The initiation of flows can then be induced by imposing constant velocity boundary conditions at the fluid inlet for instance in conjunction with periodic boundary conditions. Such settings can typically approximate the experimental practice of constant flow rates. Periodic boundary conditions (section 3.4) are particularly useful for modeling bulk systems because they tend to minimize finite size edge effects. Another important initial standard condition is the assumption of constant pressure or a pressure gradient.

While the focus in this work concentrates on slow flows through porous media the initialization method mainly applied has been the preconditioning by using a pressure gradient, whereas it is also possible to use fully developed preconditioned systems where, however, careful initialization is needed [77, 100].

4 Validation

Lattice Boltzmann implementations often use simple bounce back rules for fluid wall interaction in combination with a Single-Relaxation-Time (SRT) collision operator (BGK). These schemes are easily assembled and understood, however, it is far more efficient to use wall interface boundary conditions of higher order accuracy in combination with a Multiple-Relaxation-Time model and a calculation grid which is parameterized and decoupled from the input geometry. Due to the increased parameter space covered by more advanced models and the ability to choose the numerical resolution independently of the resolution of the input geometry, the convergence behavior can be significantly improved. This is shown in section 4.2 for the determination of saturated permeability in artificial porous media such as sphere arrays and for an application in natural porous media in section 6.3.

If a new numerical model or new extensions are introduced, even into a well known context or if an implementation has been changed it is in any event inevitable to validate model and tool. In the course of this thesis an extensive validation process has been performed and evaluated. A common method to validate the accuracy of a model is the comparison of simulation results with known properties of the system represented by that model. In order to successfully compare solutions one depends on systems where the results are known, respectively whose solutions can be derived in an analytical or, at least, in a semi analytical method. The following section presents some of the validation examples, which were carefully examined during the development of the model and code implementation.

4.1 Grid unaligned channel flow

The simple bounce back scheme for fluid wall interactions inherently assumes that the wall is exactly located in the geometric center between two grid nodes. Due to this constraint the accuracy of the whole scheme, at least for complex geometries, is limited to first order. If walls are aligned to the grid, like in channels, pipes , or the geometry is symmetric, the error in space introduced by the bounce back scheme can possibly be neglected, because the error itself is symmetric or it is taken into account, that the walls are centered between two grid nodes. Therefore, and in order to validate the higher order boundary conditions for no slip and free slip walls introduced in section 3.1 and section 3.2, it is necessary to perform experiments with unaligned walls as well as when errors are not canceled out due to a symmetric setup.

To demonstrate the accuracy of the two types, an inclined channel with an arbitrary angle has been chosen. In order to avoid the cancellation of symmetric errors, due to alignment between geometry and grid, angles of 0°, 45° or 90° should be avoided. For the numerical validation experiments channels have been created with angles of 5.71°, 16.70°, 30.96° and 41.99°. These angles cannot be chosen entirely arbitrary, because the flow field is driven by a volume force and therefore the in- and outflow at the domain boundaries must be periodic. If the outflow on the top is then copied to the bottom and the inflow vice versa, this has to be done by introducing a geometric offset, representing the dislocation of channel in- and outflow. Due to

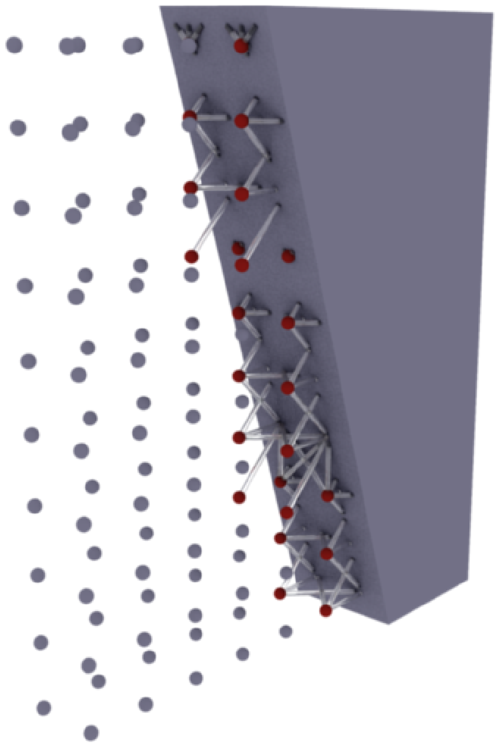


Figure 4.1.1 – Subsection of bulk fluid grid nodes and subgrid distances at the fluid/wall interface

this dislocation and the fact that the channel walls are not necessarily located in the middle between to grid points, it has to be carefully assured, that the walls are connected consistently while extrapolating the periodic setup. The setup and the constraints in choosing the angles are illustrated in Figure 4.1.2 and Figure 4.1.1.

The channel's walls have been modeled either with no-slip linearly interpolated bounce back schemes (LIBB) or with a one sided quadratically or linearly interpolated free-slip scheme. While the simple bounce back rule shows only a linear convergence behavior, the interpolated higher order boundary conditions for no slip, as well as for free slip walls, have an order of quadratic convergence as shown in Figure 4.1.3 and Figure 4.1.4.
In most applications a linear interpolated fluid wall boundary condition is sufficient to maintain a second order accurate flow field for arbitrary and even complex geometries under the assumption that accurate subgrid distances (sometimes referred to as q's) are present. The preprocessor described in section 8.1 can provide these q's from almost every geometry as long as a parameterized surface can be extracted.

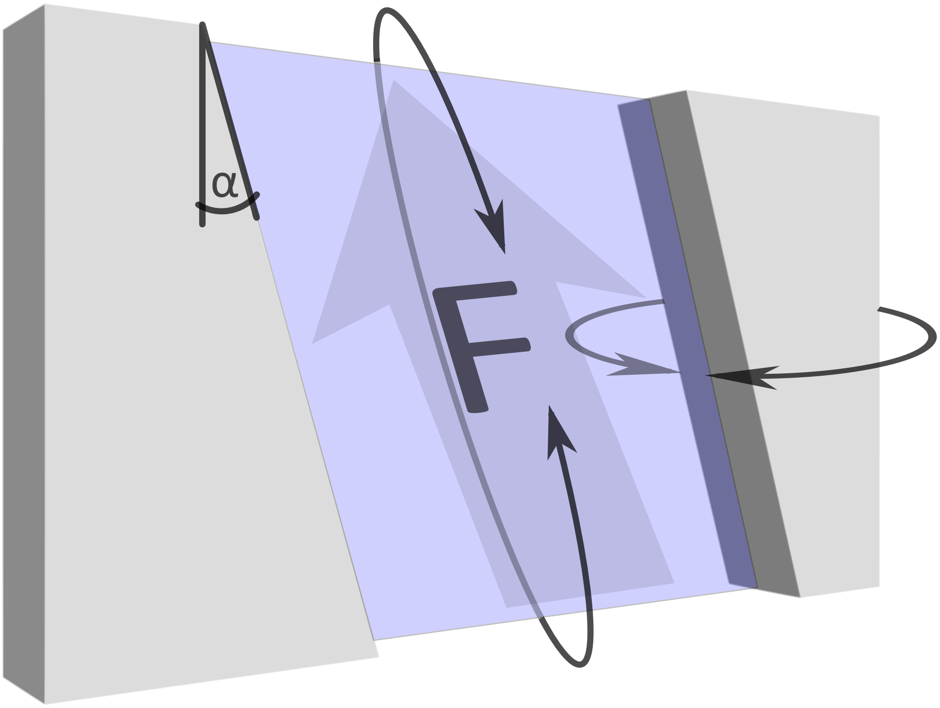


Figure 4.1.2 – Unaligned one sided free slip channel flow test case: experimental setup

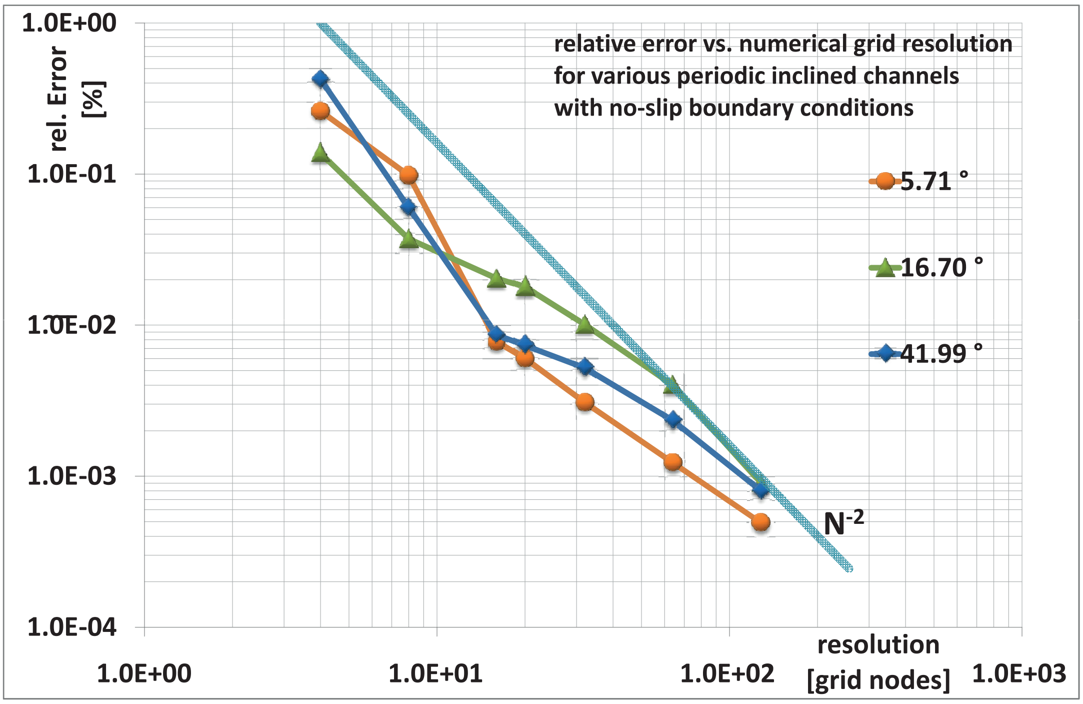


Figure 4.1.3 – Numerical results of hydrodynamic flow in an inclined channel for different angles (linear interpolated bounce back [LIBB])

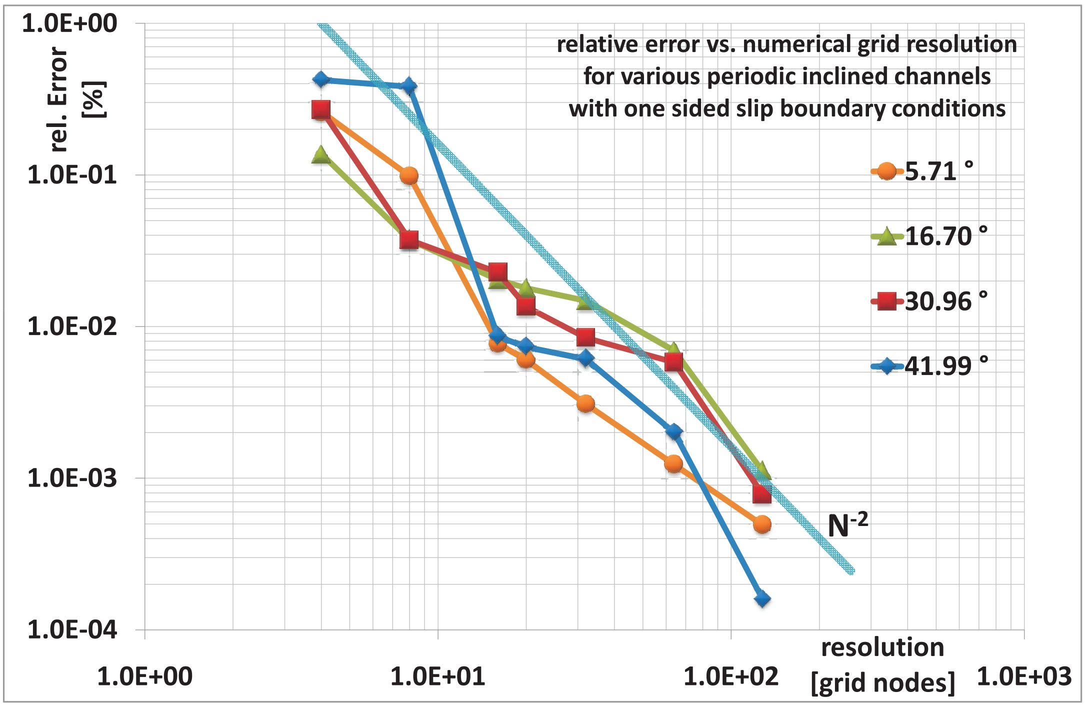


Figure 4.1.4 – Numerical results for hydrodynamic flow in an inclined channel for different angles with free-slip as well as no-slip boundary conditions (linear interpolated)

4.2 Permeability of sphere arrays

To evaluate the accuracy of the utilized method while simulating fluid flow in complex geometries it is necessary to validate the chosen approach. Furthermore, it is also important to get an idea of quality, convergence behavior, and suitability for a certain problem. For simple geometric configurations one can choose from a large variety of validation examples, like Couette- or Hagen-Poiseuille flows. These simple setups have been extensively studied and documented and the data can be compared against a solution which is analytic in nature. If the system becomes more complex and the focus is aiming on the cooperation between the grid generation process and the accuracy of interpolated fluid wall boundary conditions, suitable validation examples are hard to find. A typical validation example is the determination of the permeability of sphere arrays[1]. For a body centered cubic array of spheres (see Figure 4.2.1) one can derive a semi-analytical solution [69, 122] which has an accuracy with an error below 1% for laminar flows and furthermore in a Stokes flow regime (Re< 1) this error nearly vanishes. The force that is acting on one sphere can be determined as follows:

$$F = C_d \cdot 6\pi\mu u_d a \tag{4.2.1}$$

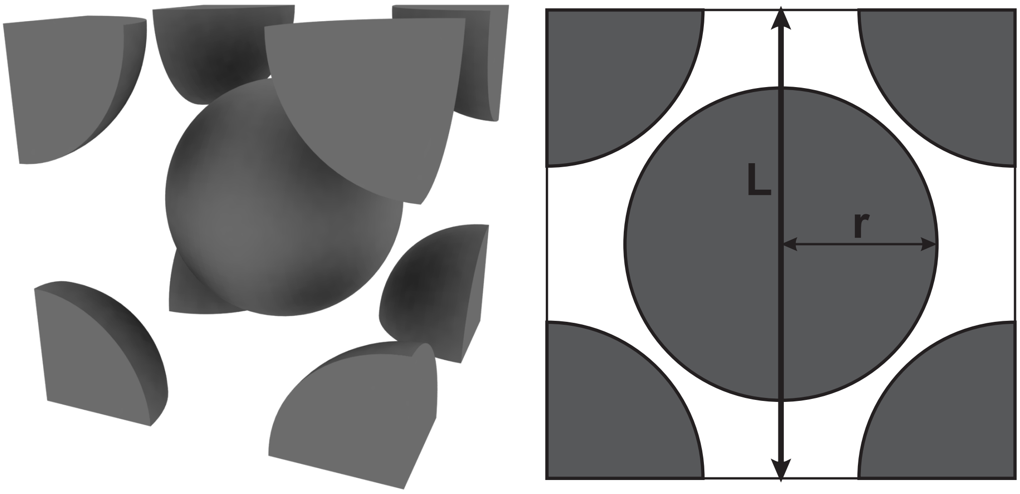
where C_d is the dimensionless drag, μ the dynamic viscosity and u_d the Darcy velocity. The drag can be computed as a function of the solid volume fraction c by a series expansion:

$$C_d = \sum_{n=0}^{30} \alpha_n \chi^n, \quad \chi = (\frac{c}{c_{max}})^{1/3}, \quad c = \frac{8\pi a^3}{3L^3}, \quad c_{max} = \frac{\sqrt{3}\pi}{8} \tag{4.2.2}$$

---

[1] permeability - Latin: permeare = to pass through, here: the ability of a material, typically a rock or an unconsolidated material, to transmit fluids

The permeability of the medium can then be derived by considering the region depicted in



(a) A BCC array of spheres in periodic space (b) 2D projection of the sphere array

Figure 4.2.1 – Validation test case for permeability measurements

Figure 4.2.1. Taking into account that the region contains two spheres in sum [3], we can compute the average pressure gradient in z-direction as

$$-\partial_z\, p = \frac{2F}{L^3}. \tag{4.2.3}$$

The Darcy velocity using equation (4.2.8) is then

$$u = \frac{\kappa}{\mu}\frac{2F}{L^3}. \tag{4.2.4}$$

Substituting Equation 4.2.1 in Equation 4.2.4 we obtain κ as the intrinsic permeability

$$\kappa = \frac{L^3}{12\pi a C_d}. \tag{4.2.5}$$

The result to compare against is the dimensionless permeability κ^*:

$$\kappa^* = \frac{\kappa}{a^2} = \frac{1}{12\pi C_d}\cdot\left(\frac{L}{a}\right)^3. \tag{4.2.6}$$

The input geometry used to generate the numerical grid on which the simulation runs, has been created using an ordinary CAD modeling tool, such as AutoCAD [10] or Microstation [14]. The triangular mesh has been exported and used as input to the grid generation tool described in chapter 5. Therefore it was easy to create grids at different resolutions using always the same input model. The smallest grid used had 10^3 nodes and the largest up to 120^3. It is also noted that the grid has been created using a slight offset compared to the triangular mesh, meaning that the grid is shifted by a fraction of an inter-node distance to avoid the cancellation of errors due to symmetry, similar to the setup used for the inclined channel in section 4.1.

As the Reynolds number of the flow is smaller than unity it is appropriate to omit the non-linear terms of the equilibrium distribution function in the MRT-model. This measure increases the numerical efficiency as well as the stability of the procedure. Furthermore this is a prerequisite for eliminating the viscosity dependency on permeability since one can deploy the magic relaxation parameters given below according to [58].

$$s_e = s_\epsilon = s_\pi = s_\nu, \quad s_q = s_m = 8\frac{(2-s_\nu)}{(8-s_\nu)} \tag{4.2.7}$$

These parameters eliminate the permeability dependence on viscosity for simulations using Stokes flow (i.e. neglecting the second order terms of the equilibrium distributions) for simple bounce-back and reduce them substantially in the case of interpolation based bounce back by relaxing the even and odd moments differently.
The permeability is then determined by measuring the volume averaged fluid velocity (Darcy velocity) after reaching a stationary state. From Darcy's law one can calculate the fluid permeability,

$$u_d = -\frac{\kappa}{\rho\nu}(\partial_z\, p + \rho g) \tag{4.2.8}$$

Those results can then be compared to the theoretical solution given by [69, 122].

Figures 4.2.2 and 4.2.3 show the comparison for different values of χ. Figure 4.2.2 reveals

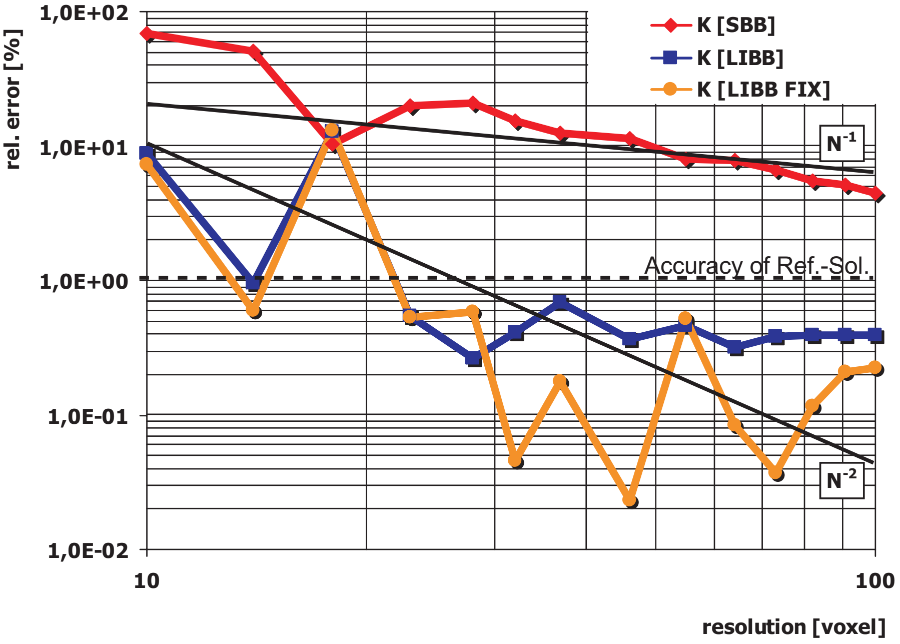


Figure 4.2.2 – Plot comparing the accuracy of simple and linear interpolated BB with $\chi = 0.76$.

that the accuracy of the simulation using the linear interpolated bounce back schemes for a resolution of about 14^3 grid nodes is comparable to the one from the truncated series expansion of the theoretical solution (relative error below 1%). Additionally, it is noted that the order of convergence for the simple bounce back scheme is always linear, whereas the order is quadratic for linear interpolated bounce back.

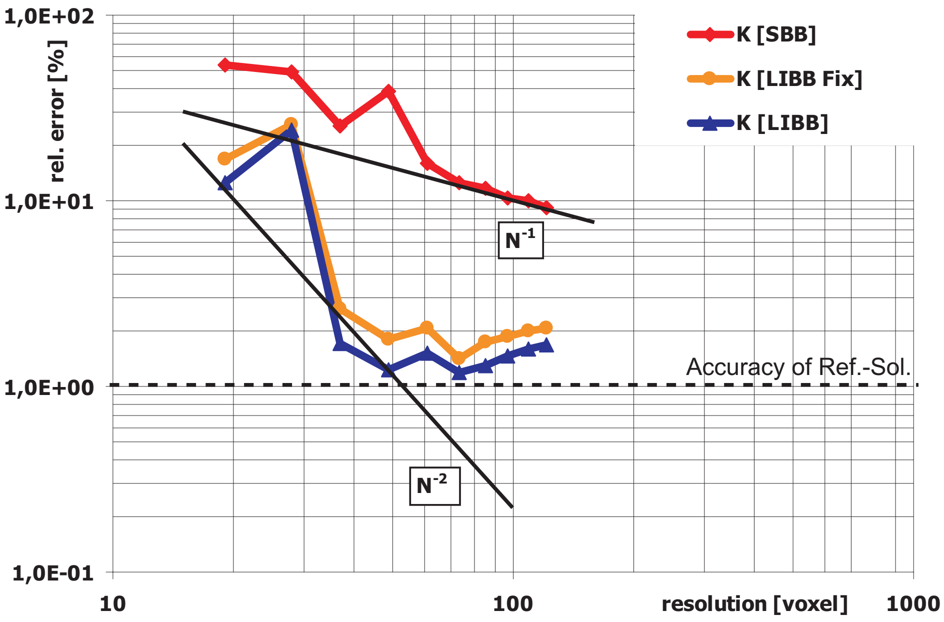


Figure 4.2.3 – Plot comparing the accuracy of simple and linear interpolated BB with $\chi = 0.96$.

4.3 Drag on spheres

In order to further validate the free slip boundary condition described in section 3.2 and as a prerequisite for the substitution of fluid-fluid interfaces against free slip walls, experiments determining the drag on solid and fluid spheres have been performed. The experimental setup has been kept simple: a circular tube forms the fluid domain and a solid sphere with either no-slip or free-slip boundary conditions has been located at the middle axis of the tube, separating the first third from the second two thirds of the pipe length. Velocity boundary conditions have been applied to the inflow and outflow area as well as to the tube walls to imitate a moving sphere in a pipe (see also Figure 4.3.2). The drag force on the obstacle induced by the flow field can be computed by the method of momentum exchange as described by [111] or by stress integration over the obstacle as proposed by [105].
The force $\mathbf{F}_k$ acting on a boundary cut by a link k between $\mathbf{x}_f$ and $\mathbf{x}_b$ results from the momentum-exchange between the mass fraction $f_{\hat{i}}(t, \mathbf{x}_f)$ and $f_i(t + \Delta t, \mathbf{x}_f)$ hitting the boundary as shown in Figure 4.3.1. The momentum change can be computed by regarding the mass fraction before and after hitting the boundary:

$$\mathbf{F}_k(t + \Delta t/2) = -\frac{V}{\Delta t}\mathbf{e}_i \left(f_i(t + \Delta t, \mathbf{x}_f) + f_{\hat{i}}(t, \mathbf{x}_f) \right), \tag{4.3.1}$$

where V is the volume of the unit cell. Drag and lift forces on the whole obstacle are computed by summing up all contributions $\mathbf{F}_k$,

$$\mathbf{F} = \sum_{k \in \mathfrak{C}} \mathbf{F}_k, \tag{4.3.2}$$

where $\mathfrak{C}$ is the set of all links cut by the obstacle and the sum considers only boundary nodes $\mathbf{x}_f$. A detailed comparison of both methods, momentum exchange as well as stress integration, can be found in [105]. In this numerical experiment the method of momentum exchange has been used and, for this type of problem, can be considered as sufficiently accurate.

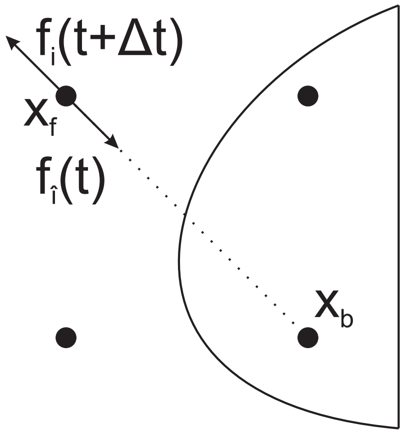


Figure 4.3.1 – Momentum transfer on fixed obstacles

4.3.1 A solid moving sphere in a pipe at Re=1

An approximate solution [125] for the dimensionless drag coefficient for a moving sphere with speed U_0 in an infinite fluid is given by

$$c_d = \frac{24}{\mathrm{Re}}\,(1 + 0.15\,\mathrm{Re}^{0.687}). \tag{4.3.3}$$

The Reynolds number is defined as

$$\mathrm{Re} = \frac{U_0\, d}{\nu}, \tag{4.3.4}$$

where d is the diameter of the sphere and ν the kinematic viscosity. The drag force F_d exerted on the sphere is

$$F_d = c_d\, \frac{1}{2}\, \rho_0\, U_0^2\, \pi\, \frac{d^2}{4}. \tag{4.3.5}$$

The relative error of the approximation given in Equation 4.3.3 is ± 5 % for $\mathrm{Re} < 800$. For a moving sphere in an infinite pipe the influence of the wall can be taken into account [44] as

$$c_{d,W50} = c_d + \frac{24}{Re}(K-1), \tag{4.3.6}$$

where K is given by [67]

$$K = \frac{1 - 0.75857\,\lambda^5}{1 - 2.1050\,\lambda + 2.0865\,\lambda^3 - 1.7068\,\lambda^5 + 0.72603\,\lambda^6} \tag{4.3.7}$$

and $\lambda = \frac{d}{D}$ is the ratio of the diameters of the sphere and pipe. Approximation (4.3.6) has a relative error of± 5 % for $\mathrm{Re} < 50$ and $\lambda < 0.6$. In the range $100 < \mathrm{Re} < 800$ the dimensionless drag coefficient is given by

$$c_{d,W800} = k_f\, c_d, \tag{4.3.8}$$

where k_f is given by [34]

$$k_f = \frac{1}{1 - 1.6\,\lambda^{1.6}}. \tag{4.3.9}$$

Approximation (4.3.8) has a relative error of ± 6 % for $\lambda < 0.6$.

The reference frame is a coordinate system moving with the sphere, leading to a setup as shown in Figure 4.3.2. No-slip conditions are imposed on the boundary of the sphere and velocity boundary conditions on the inflow and outflow as well as on the boundary of the pipe. The moving sphere simulations have been performed for a Reynolds number smaller than unity and at three different resolutions. While increasing the mesh resolution, the Mach number has simultaneously been reduced by lowering u_0. The force on the sphere is computed using equation (4.3.2) and the drag coefficient is obtained from equation (4.3.5). The reference values for the drag coefficient is $c_{d,W} = 144.48$. In Table 4.3.1 the results are given and one can clearly observe a convergent behavior for $c_{d,W}$ with an increasing mesh resolution.

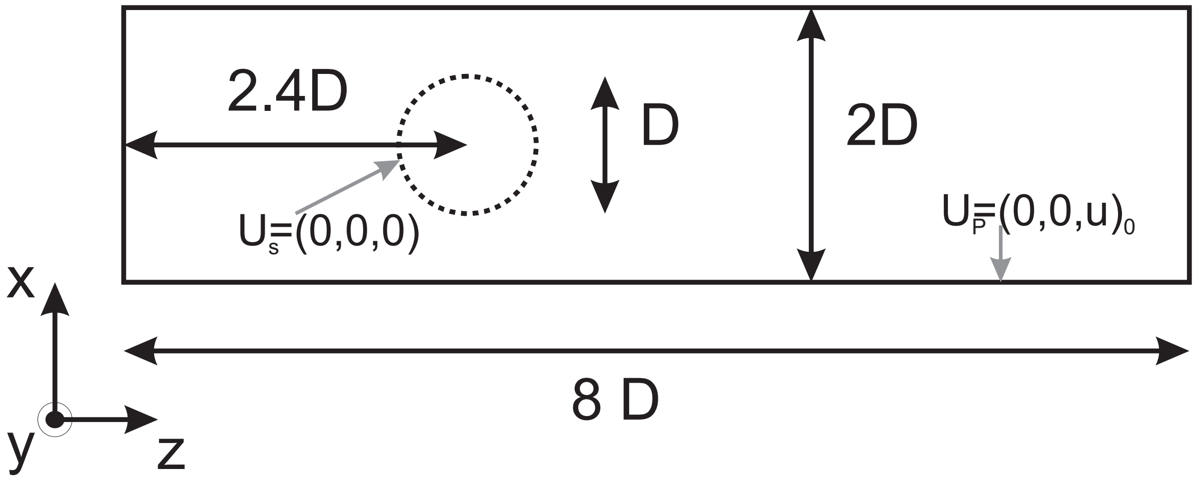


Figure 4.3.2 – Moving sphere in a pipe, setup for numerical simulation

Table 4.3.1 – Moving sphere at Re<1, relative errors for different mesh sizes

domain size	u_0 [m s^{-1}]	ν [m^2 s^{-1}]	D [m]	$c_{d,W}$ [-]	Rel. Err. [-]
$32^2 \times 128$	0.004	0.0595	14.88	153.1	5.6 %
$64^2 \times 256$	0.002	0.0605	30.24	146.9	1.6 %
$128^2 \times 512$	0.001	0.0610	60.96	145.4	0.6 %

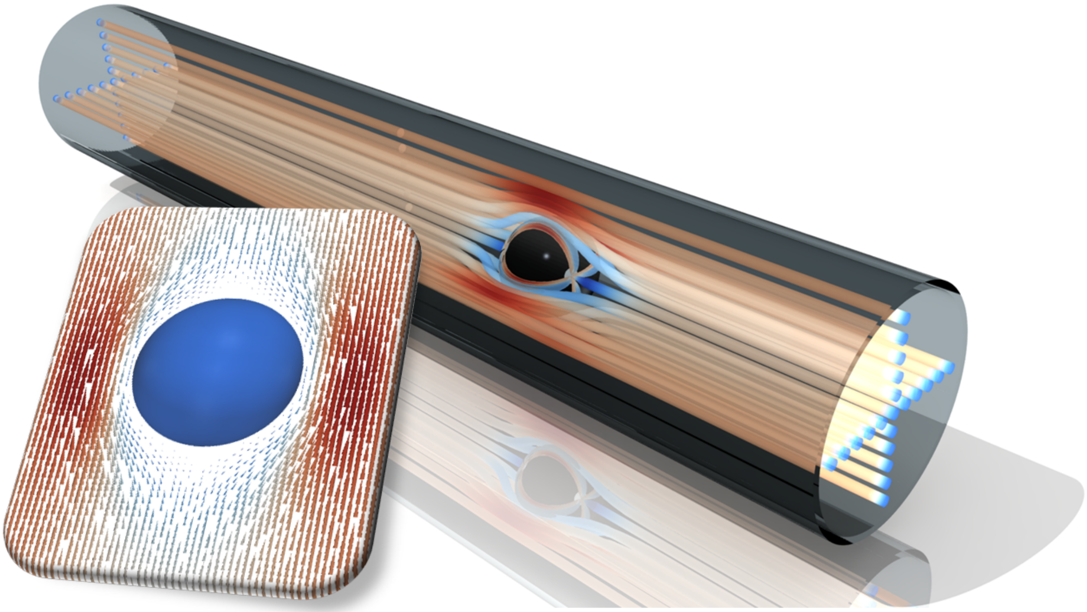


Figure 4.3.3 – Streamlines over a solid sphere (no-slip LIBB)

4.3.2 A liquid sphere moving in a pipe at Re=1

The experiment described in detail in the preceding subsection 4.3.1 has been repeated using the same setup and parameter space, except the fluid/wall boundary conditions around the sphere have been changed to free-slip type mimicking a liquid sphere. The drag force then computes as:

$$F_d = 2\,\frac{\pi\,\nu\,r U_0 K\,(2+3\,\lambda)}{1+\lambda} \tag{4.3.10}$$

and the correction factor K of the wall influence changes to the one given in equation 4.3.11.

$$K = \left(1 + 2.2757\,\frac{\beta^2\,(1-\lambda)}{2+3\,\lambda}\right) \left(1 - 0.7017\,\frac{(2+3\,\lambda)\,\beta}{1+\lambda} + 2.0865\,\frac{\lambda\,\beta^3}{1+\lambda} + 0.5689\,\frac{(2-3\,\lambda)\,\beta^5}{1+\lambda} - 0.72603\,\frac{(1-\lambda)\,\beta^6}{1+\lambda}\right)^{-1} \tag{4.3.11}$$

The drag coefficient on a liquid and moving sphere computes similar to the one of a rising bubble [34]:

$$C_d = 2\,\frac{Fd}{U^2\pi\,r^2} \tag{4.3.12}$$

Also during this numerical experiment the mesh resolution has been increased while the Mach number was simultaneously reduced by lowering u_0. The force on the sphere is computed using Equation 4.3.10 and the drag coefficient is obtained from Equation 4.3.12. Here, the reference value for the drag coefficient is $c_d = 51.37$.

The results given in Table 4.3.2 are similar to the experiment performed with the solid sphere before and one can clearly observe a convergent behavior for c_d with increasing mesh resolution.

In summary, both types of fluid wall boundary conditions show the expected behavior (see Figure 4.3.5): In contrary to a simple bounce back scheme, the approach of interpolating the distributions dependent on the subgrid distances leads to second order accurate results. Therefore, for complex geometries the precision of the whole model is not degraded by the boundary conditions. However, it has to be mentioned that interpolated bounce back schemes, independently of their polynomial order (linear, quadratic or higher order) are not mass conservative. To circumvent this problem the mass difference can e.g. be transfered to the rest particle distribution f_0 [5]. This results in an interpolated bounce back scheme which is conservative in mass (and thus pressure), but introduces a higher-order disturbance of the stress tensor. However,

Table 4.3.2 – Moving liquid sphere at Re<1, relative errors for different mesh sizes

domain size	u_0 $[\mathrm{m\,s^{-1}}]$	ν $[\mathrm{m^2\,s^{-1}}]$	D [m]	$c_{d,W}$ [-]	Rel. Err. [-]
$32^2 \times 128$	0.004	0.0595	14.88	61.2	16.1 %
$64^2 \times 256$	0.002	0.0605	30.24	55.1	6.8 %
$128^2 \times 512$	0.001	0.0610	60.96	53.2	3.4 %

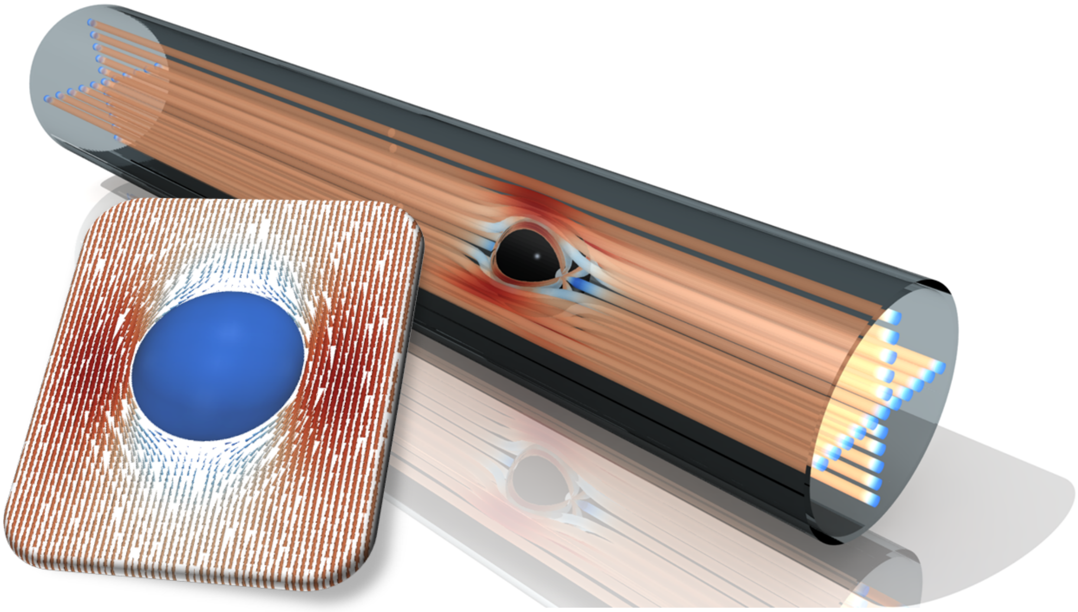


Figure 4.3.4 – Streamlines over a fluid sphere (Slip LIBB).

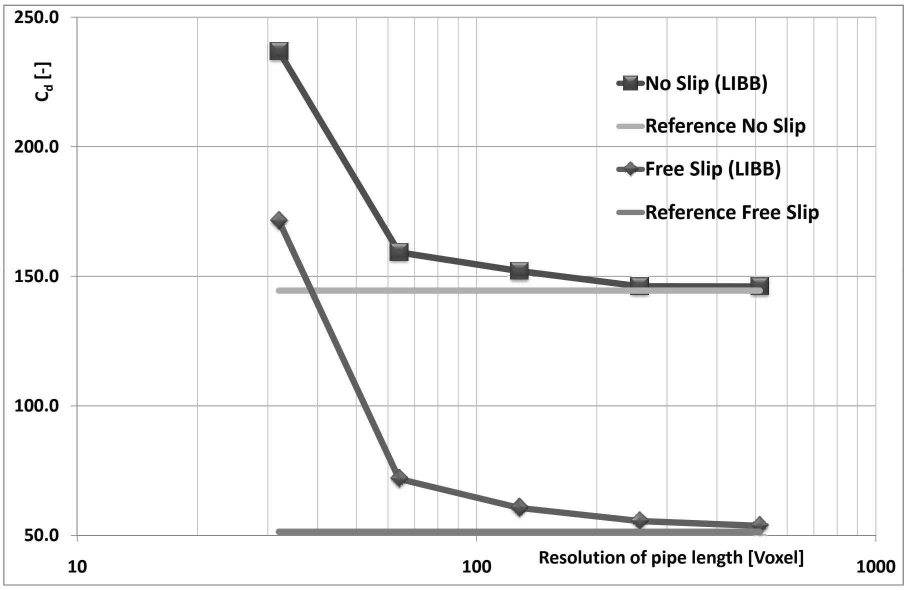


Figure 4.3.5 – Comparison of the drag force of the numerical experiments vs. analytical results, either for no slip and for slip boundary conditions.

this method does not change the results significantly and guarantees mass conservation.

Even the most sophisticated type of wall treatment, the class of multi-reflection boundary conditions [60], are, besides their unmatched accuracy, not mass conservative. Multi-reflection boundary conditions have not been implemented in this branch of VirtualFluids Classic, because linear or quadratic interpolated bounce back schemes are considered sufficiently accurate for very low Reynolds numbers which is typical for fluid flow in porous media.

4.4 Bubbles and Droplets

While the preceding sections verified the validity of boundary conditions for single phase flow the subsequent sections indicate the accuracy and the physical quality of the chosen multiphase approach introduced in section 2.2. The first validation example determines the properties of the model extension in an unbound fluid with no fluid-wall interactions. By placing a bubble of a non wetting fluid into an unbound domain consisting of a wetting fluid and by applying surface tension to the fluid-fluid interface it can be shown that the Young-Laplace law is fulfilled. The Young-Laplace equation is a nonlinear partial differential equation that describes the capillary pressure difference, Δp, sustained across the interface between two static fluids, such as water and air, due to the phenomenon of surface tension:

$$\begin{aligned}\Delta p &= \sigma \nabla \hat{n}, &(4.4.1)\\ &= 2\sigma H, &(4.4.2)\\ &= \sigma \left\{ \frac{1}{R_1} + \frac{1}{R_2} \right\} &(4.4.3)\end{aligned}$$

Where σ is the surface tension, $\hat{n}$ is a unit normal to the surface, H is the mean curvature, and R_1 and R_2 are the principal radii of curvature. If this law is applied to a bubble or droplet in a static fluid with no motion of neither the bubble nor the surrounding fluid and accordingly the mean curvature is constant, one can conclude that the pressure difference between the two phases can be written as follows:

$$\Delta p = \frac{2\sigma}{r}, \tag{4.4.4}$$

where r is the radius of the bubble. Simulations that have been carried out showed that the agreement in pressure difference, Δp, between the lattice Boltzmann simulations and the theoretically predicted values is excellent (Table 4.4.1).

Extensive theoretical and experimental works on rising bubbles are found in the books of [34] and [166]. In the case of a free rising bubble in an infinite medium, the bubble motion can be

Table 4.4.1 – Accordance of Young-Laplace Law between LB-simulations and analytically predicted values

Radius	σ	$\Delta p_{Laplace}$	$\Delta p_{measured}$	rel. error
5	0.01	0.004	0.00423	5.75
10	0.01	0.002	0.00205	2.5
20	0.01	0.001	0.00099	1.0

characterized by the viscosity ratio γ, the Eötvös number Eo, the Morton number M and the Reynolds number Re [34]:

$$\gamma = \frac{\mu_i}{\mu}, \quad \mathrm{Eo} = \frac{g\Delta\rho d^2}{\sigma}, \quad \mathrm{M} = \frac{g\mu^4\Delta\rho}{\rho^2\sigma^3} \quad \mathrm{Re} = \frac{U_T d\rho}{\mu}, \tag{4.4.5}$$

where μ_i and μ are the dynamic viscosity of the inner and outer phase, $\Delta\rho$ the density difference, d the sphere diameter, ρ the density of the outer phase, U_T the terminal velocity, and g the acceleration due to gravity. The shape of the bubble can be spherical, ellipsoidal, “skirted”, “dimpled” or a “spherical cap”, depending on these parameters. When $\mathrm{Eo} > 40$ and $\mathrm{M} > 200$ the terminal velocity of a rising bubble can be computed by solving the following equation for the Reynolds number Re [34]:

$$2\mathrm{Re}^2 + 6\mathrm{Re}\frac{2+3\gamma}{1+\gamma} - \mathrm{Eo}^{\frac{3}{2}}\mathrm{M}^{-\frac{1}{2}} = 0 \tag{4.4.6}$$

The experimental results on rising bubbles, showing a good agreement, can be found in Table 4.4.2.

To further demonstrate the suitability and the possible *parameter range* covered by the chosen multiphase extension, various numerical experiments have been carried out. One is a Rayleigh-Taylor instability, or RT instability (after Lord Rayleigh [118] and G. I. Taylor [141]), which is an instability of an interface between two fluids of different densities, which occurs when the lighter fluid is pushing the heavier fluid and vice versa. This process is evident not only in many terrestrial examples, from salt domes to weather inversions, but also in astrophysics and (electro-)hydrodynamics. Here, an instability of multiple modes has been computed in parallel at a grid size of 300 × 300 × 150 resulting in 13.500.000 cells. The viscosity ratio has been chosen to 1 : 100, the density ratio to 10 : 1 and the surface tension to 0.001. A snapshot of the simulation is shown in Figure 4.4.3.

Table 4.4.2 – Comparison of predicted terminal velocity of a rising bubble and numerical experiment (Gravity set to $1.5E-5$ for the numerical experiments).

Resolution	Ratio of $\frac{\rho_W}{\rho_N W}$	r_{bubble}	r_{tube}	Eo	M	Re	U_t	U_{sim}
$64^2 \times 512$	5 : 1	30	62	1260	$350E3$	3.06	0.017	0.015
$128^2 \times 1024$	5 : 1	62	126	1260	$350E3$	3.87	0.022	0.024

It is noted that *density ratios* of 1 : 10 are easily possible, but ratios larger than 1 : 50 are commonly unstable. The *ratio of viscosity*, however, can be up to $1 : 1E06$ assuming the MRT-model or an even more sophisticated model is used and no *surface tension* is applied.

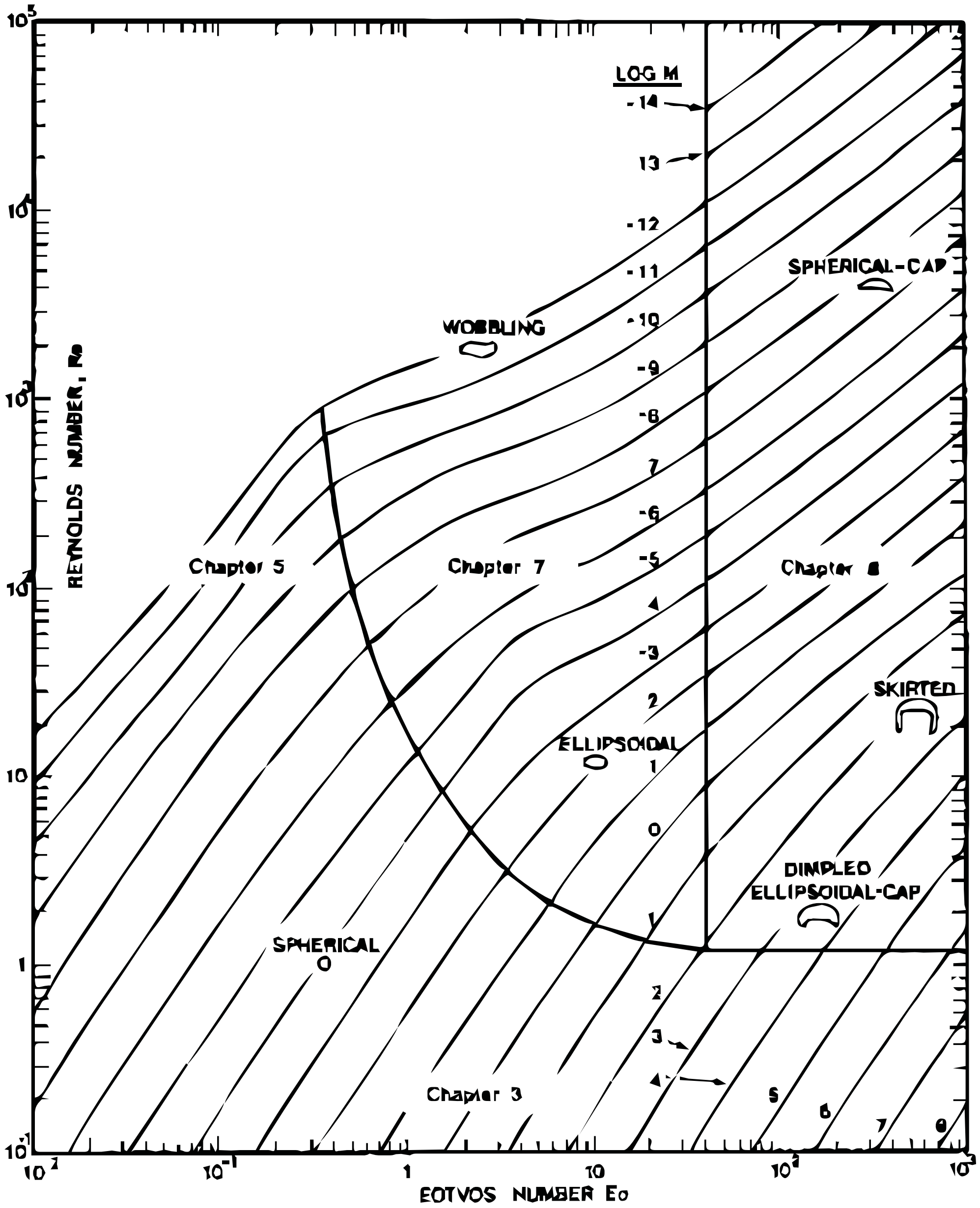


Figure 4.4.1 – Dependency of shape regimes from Reynolds-, Eotvös- and Morton-Number.

4.5 Contact Angles

Section 4.4 shows the ability of the LBM multiphase extension introduced in section 2.2 to handle two phase flow problems. However, fluid simulations in three phase contexts are far more challenging, because additional physical parameters have to be considered. One of them is the contact angle or triple point between non-wetting, wetting and solid phase (Figure 4.5.1).

Due to the chosen multiphase approach one has the freedom to arbitrary select the wetting properties of the solid phase by setting the ordering parameter Θ to a value from -1 (non-wetting/hydrophobic) to 1 (wetting/hydrophilic) or somewhere in between. Simulations with

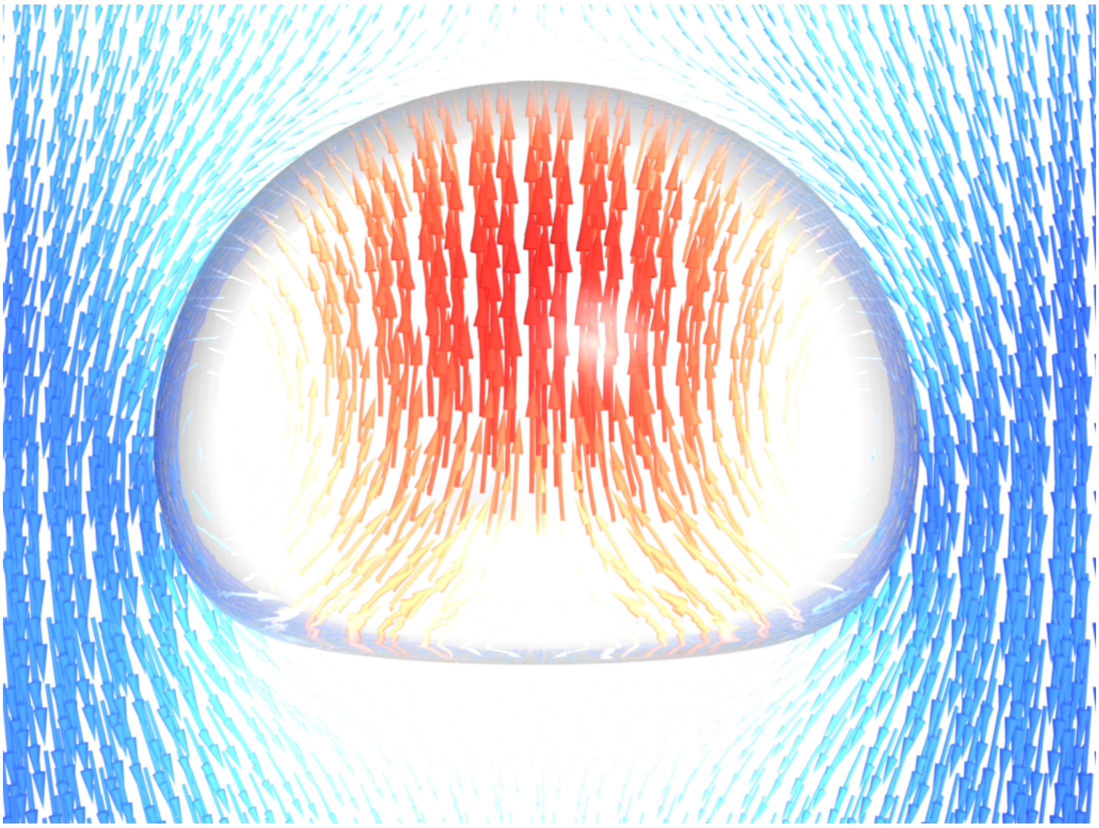


Figure 4.4.2 – rising bubble at terminal velocity with a slice showing the local velocity and developed Hill eddies.

the lattice Boltzmann model used, showed that the static contact angle are in a very good agreement to the values analytically predicted. Despite an accurate static contact angle, it is crucial to evaluate the properties of the model when the interface moves, because the contact angle depends linearly, at least at low flow rates, on the motion of the contact line. The dynamic contact angle is one of the reasons for the hysteretic behavior of the capillary pressure saturation relationship in porous media which are an important quantity in this thesis. Because the contact angle is very important for simulations of hysteretic effects in porous media it is mandatory to evaluate the contact angle properties for dynamic setups with care. Using the MRT-model multiphase extension different numerical validation experiments have been performed. While using this model it is possible to choose its hydrophilic or hydrophobic property arbitrarily for the solid phase, where one of the liquid phases has a completely wetting and the other has a dewetting property. The most comprehensive experiment, which is a droplet on an inclined surface, is briefly introduced. The typical setup for that kind of experiment is as follows: A droplet of water slips down on an inclined surface of 45 degrees driven by gravity. The experiment has been repeated with different Forcings resulting in different terminal velocities and receding as well preceding contact angles (see Figure 4.5.3). The material properties for the solid phase, ranging from hydrophilic to hydrophobic, have been considered as well. The simulation results show a good agreement to the analytical prediction as shown in Table 4.5.1. Therefore, no additional modeling of the dynamic contact angle is necessary for the problems considered in

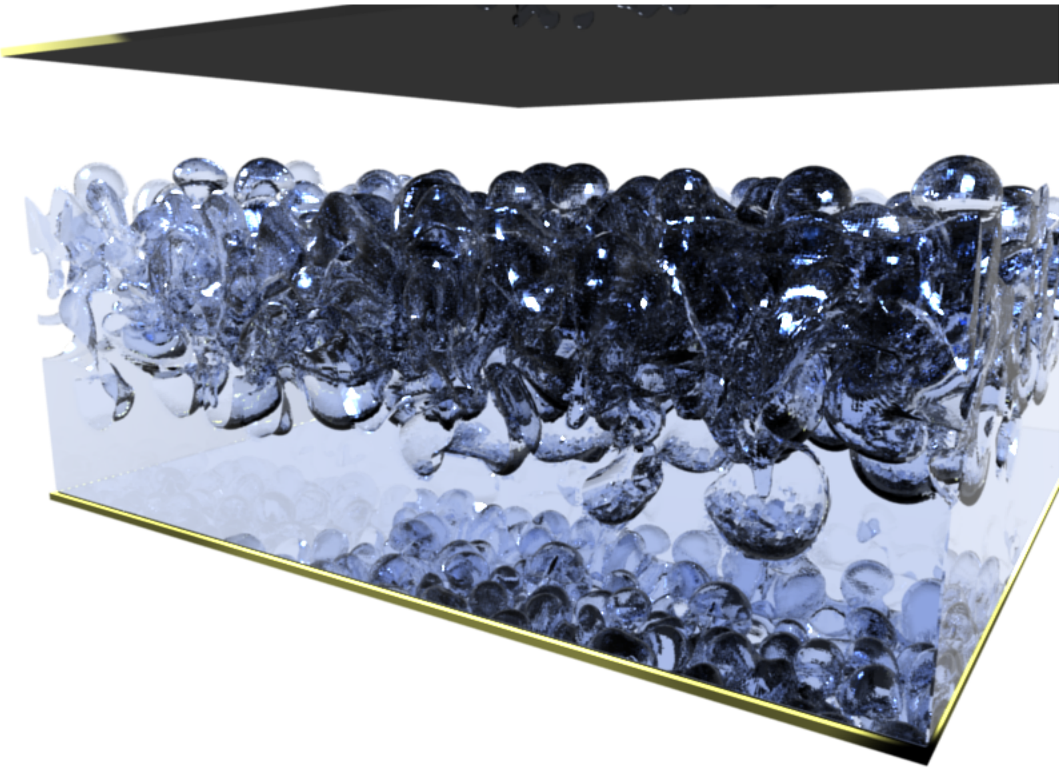


Figure 4.4.3 – 3D fingering in a Rayleigh-Taylor instability

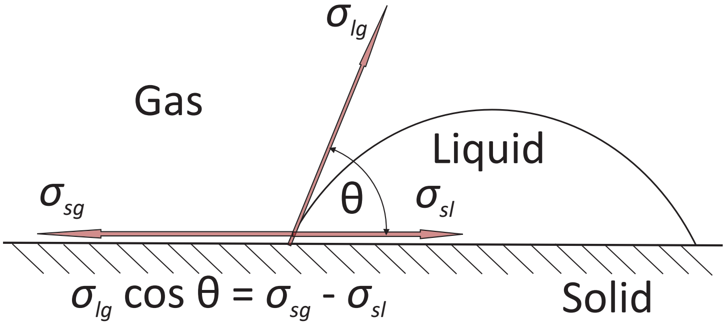


Figure 4.5.1 – Contact angle Θ of a wetting liquid on a non wetting surface

this thesis.

4.6 Pendular rings

The shape of fluid bridges between spheres have been derived using the Laplace equation by [128]. The influence of gravity was neglected and capillary pressure saturation relationships in

Table 4.5.1 – Agreement of the contact angle of a droplet on an inclined plane to predicted results

Radius	σ	$\Delta p_{Laplace}$	$\Delta p_{measured}$	rel. error [%]
5	0.01	0.004	0.00391	2.25
10	0.01	0.002	0.00198	1.00
20	0.01	0.001	0.001003	0.30
5	0.1	0.04	0.0413	3.25
10	0.1	0.02	0.0203	1.50
20	0.1	0.01	0.0099	1.00

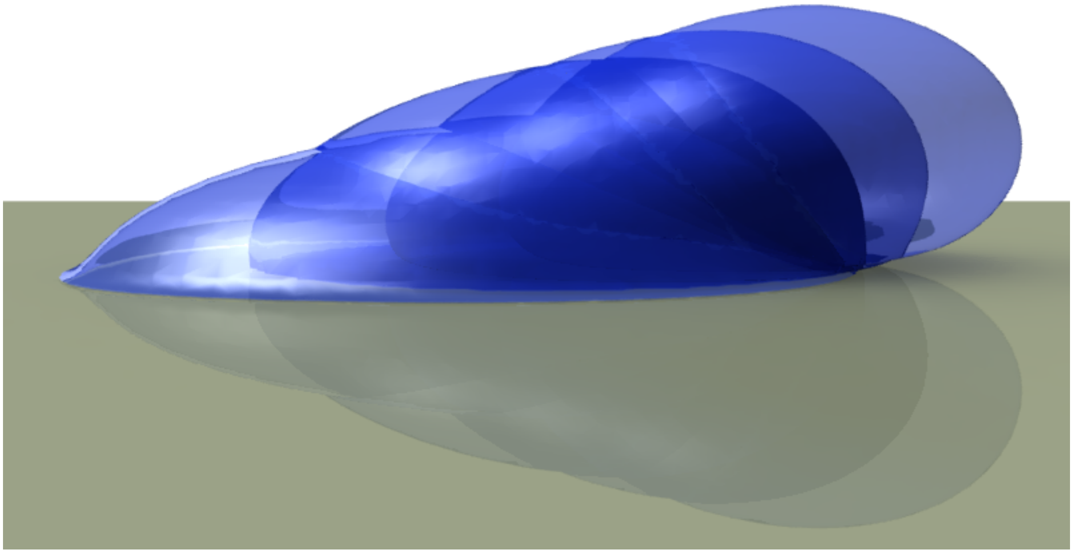


Figure 4.5.2 – overlaying droplets representing different material properties of the surface, from non wetting to wetting.

sphere packings, assuming that the wetting phase is distributed as pendular rings, have been tabulated ([128], see page 128, Figure 4.12 and page 128, Figure 4.24 or Figure 4.6.2). The values can be recovered from the tables with an accuracy of approximately 5 %. As a test case, a Body Centered Cubic (BCC) array of spheres is used (Figure 4.6.1). The porosity has been adjusted to 0.32 and periodic boundary conditions for all directions are imposed and the wetting phase is pre-initialized in such a way that pendular rings result as a static solution. The capillary pressure is computed by averaging the pressures in both phases and computing the difference. The parameters for the simulation have been chosen as follows: The grid resolution is 96^3, the viscosities are $\mu_w = \frac{1}{60}$ [Pa s], $\mu_n = \frac{1}{600}$[Pa s] and the interfacial tension is $\sigma = 0.01$[N m^{-1}]. These values are close to the values used for the simulations in a natural porous medium carried out in section 6.7. The results are shown in Figure 4.6.3, where the saturation against the

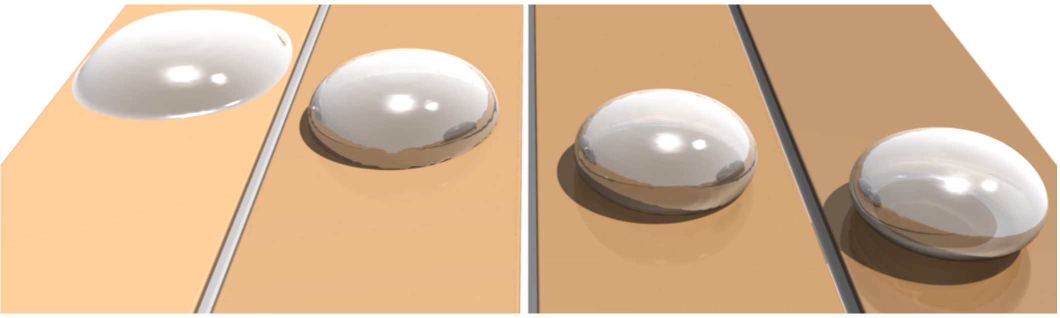


Figure 4.5.3 – droplets representing different material properties of the surface, from wetting on the left to non wetting on the right.

dimensionless capillary pressure $p_c\, d/\sigma$ (d is the diameter of the sphere) is shown. In the case of low saturations, the wetting phase is only resolved by a few grid points, therefore, the results for the capillary pressure partly deviates from the results given in [128] due to insufficient numerical resolution. In Figure 4.6.1 the resulting shape for two configurations are shown.

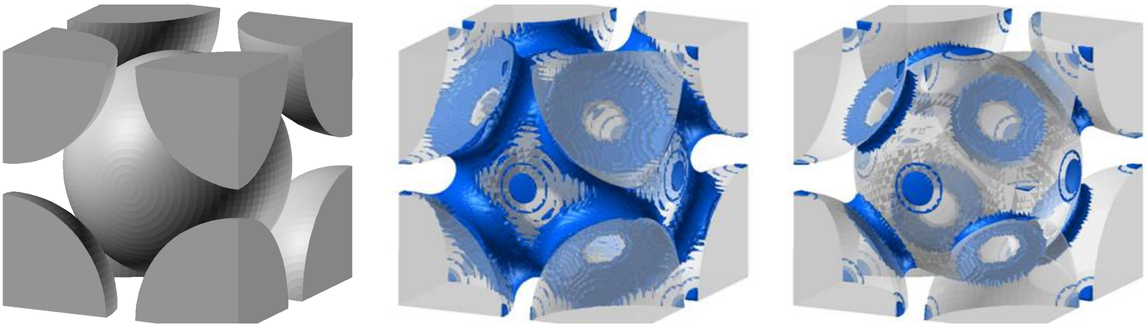


Figure 4.6.1 – Pendular rings between solids at different saturations in a BCC array of spheres. Left: geometry, middle & right: computational results

4.7 Static capillary tube bundle

As a test case, the capillary pressure saturation relationship for a bundle of capillary tubes is computed by a numerical simulation. The layout of the tube bundle is shown in Figure 4.7.1. The radii differ by one order of magnitude; the diameter of the smallest tube is resolved by approximately 4 grid points, the largest by approximately 20 grid points. The overall system is resolved by 124×108 grid points in the cross-section and 100 grid points along the tubes. The viscosities are $\mu_w = \frac{1}{60}$ [Pa s], $\mu_n = \frac{1}{600}$ [Pa s] and the interfacial tension is $\sigma = 0.01$ [N m^{-1}]. The system has been initialized with different saturations where the wetting phase was distributed at a certain fill level uniformly over the different tubes. The tubes are connected at the top to the non-wetting phase reservoir and at the bottom they are interconnected by a small layer of 2 grid spacings perpendicular to the tubes. Thus, the system is allowed to equilibrate, so that the largest tubes are evacuated and the smaller ones are filled. After reaching a steady state the

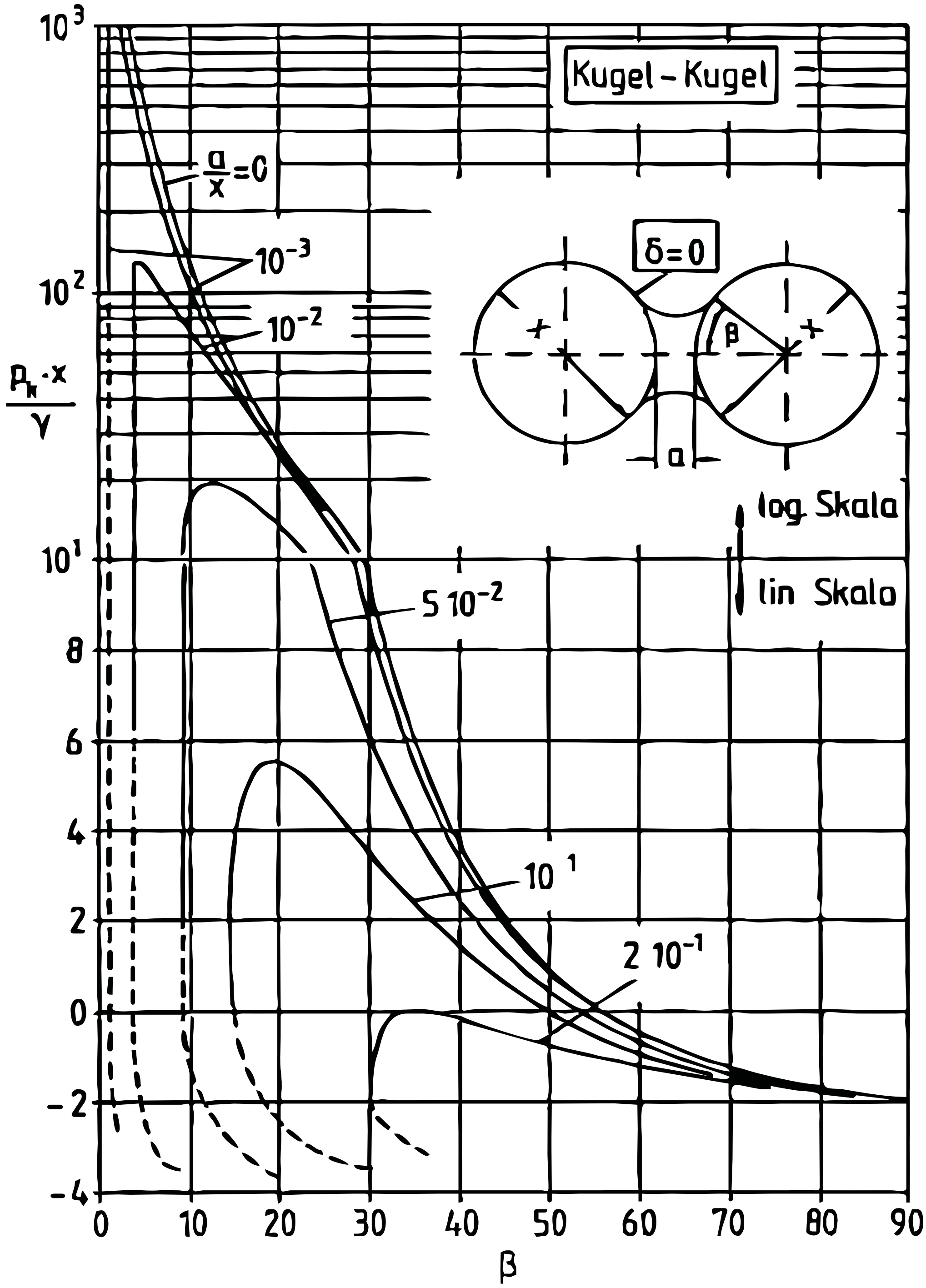


Figure 4.6.2 – Capillary pressure for pendular rings between two spheres at a contact angle of 0°

pressure is measured in the wetting phase. The theoretical solution is piecewise constant and compared to the results of the simulations represented by dots in Figure 4.7.1. The pressure

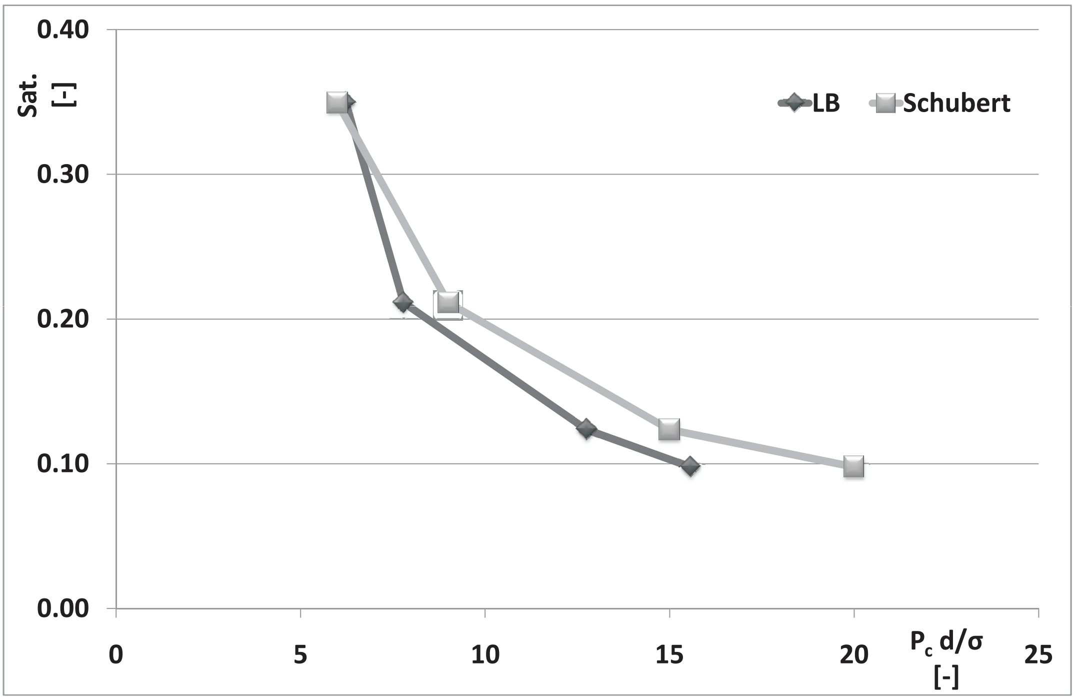
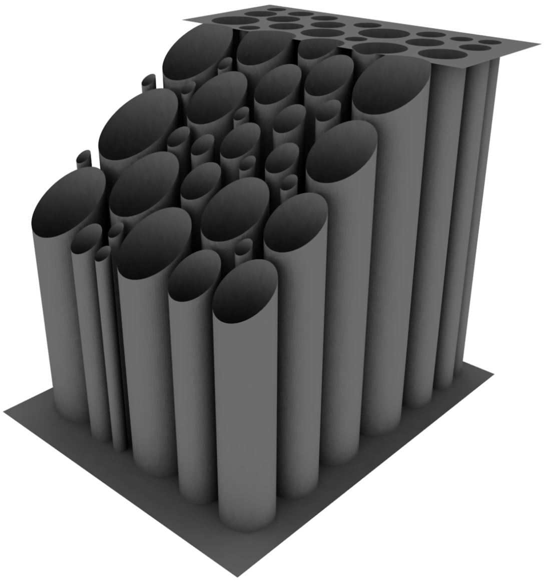


Figure 4.6.3 – Capillary pressure for pendular rings at different saturations

is normalized with respect to the entry pressure $2\,\sigma\,/\,r_{max}$. In the case of low saturations the wetting phase is only resolved by a few grid points, so that the numerical model again suffers from insufficient resolution and the resulting capillary pressure, somewhat, deviates from the analytical results.



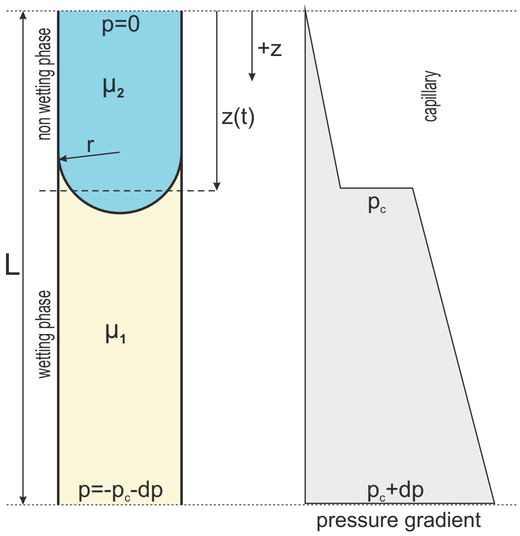


Figure 4.7.1 – right: static capillary tube bundle, left: Definitions for the Washburn equation

4.8 Capillary dynamics

The rise of a single capillary tube as shown in Figure 4.8.1, driven by a pressure difference dp at the top and the bottom can be computed by the Washburn solution [159] in the absence of gravity according to

$$z(t) = \frac{1}{\mu_w - \mu_n}\left(\mu_w\, L - \sqrt{\left(L^2\mu_w^2 + \frac{dp\, r^2}{4}\,(\mu_n - \mu_w)\; t\right)}\right). \tag{4.8.1}$$

For the simulation the ratio of the length L and the radius r is $L/r = 20$. We have performed simulations with two different setups: In the first simulation both phases had the same viscosity $\mu_w = \mu_n = 1/6$ [Pa · s], in the second simulation the ratio of viscosities has been set to 10, $\mu_w = 1/6$ [Pa · s], $\mu_n = 1/60$ [Pa · s]. The surface tension was set to 0.01 [N m^{-1}] and the driving pressure was $dp = 1.30\; p_c$, where $p_c = 2\,\sigma\,/\,r$. In Figure 4.8.2 the results for both cases are given, whereas the height is *height* $= z(t)/L$ and the x-axis presenting the dimensionless time $T\,\sigma\,/\,(L\,\mu_w)$.

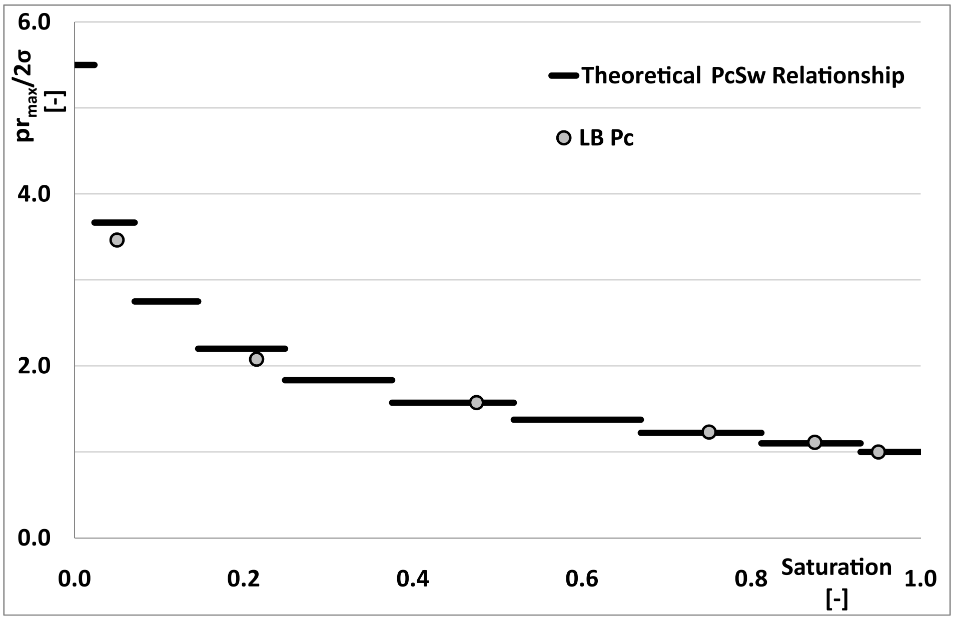


Figure 4.8.1 – capillary pressure - saturation relationship for a capillary tube bundle

4.9 Dynamics of a capillary tube bundle

The tube bundle setup discussed in section 4.7 has been used for different single step outflow experiments. The grid resolution and the material parameters are the same as in section 4.7

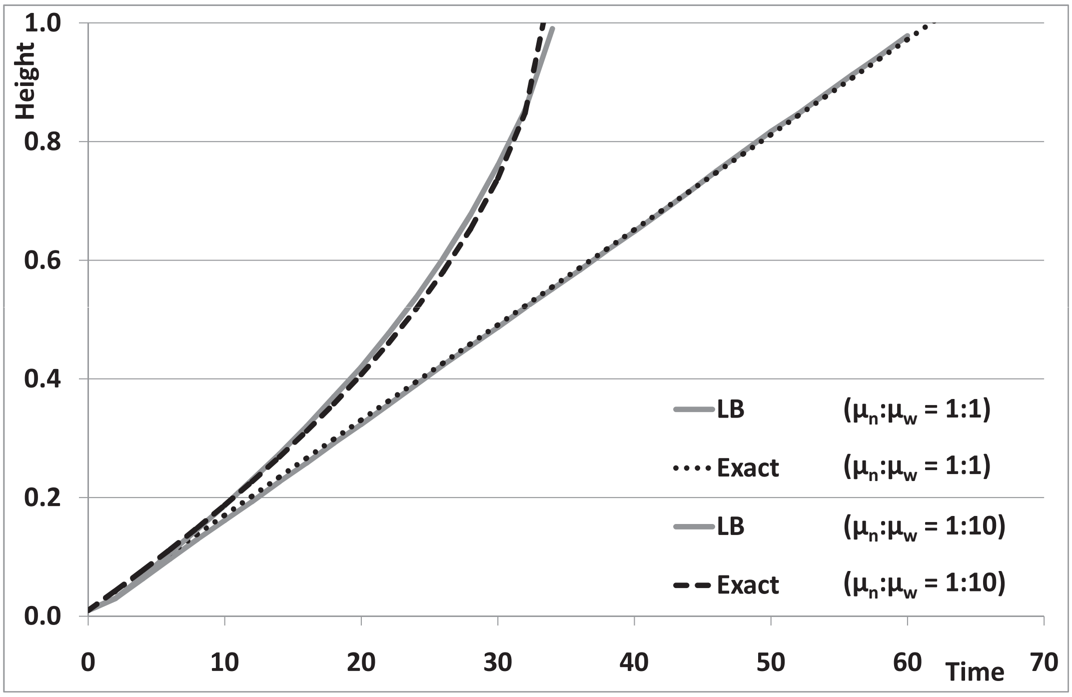


Figure 4.8.2 – Results from the numerical capillary rise experiment

and the initial velocity has been set to $\mathbf{u}_0 = 0$. Three simulations with different pressure loads P1, P2, P3 shown in Figure 4.9.1 have been performed. The analytical solution of the outflow of the wetting phase can be computed by integrating the Washburn solution (Equation 4.8.1)

$$V(t) = \pi \sum_{k=1}^{M} N_k r_k^2 (L - z_k(t)), \tag{4.9.1}$$

where M is the number of different classes of tubes, N_k is the number of tubes within one class, r_k is the radius and $z_k(t)$ is the Washburn solution (Equation 4.8.1). In Figure 4.9.2 the results from the simulation and the analytical solution are given for the three different test cases. The time scale has been normalized with respect to the time T_e, where the analytic solution does not change anymore. Besides minor deviations at the start and end of the drainage process, where the assumption for the Washburn equation are only approximately fulfilled, the agreement is good.

4.10 Residual water saturation in a porous medium

Due to the phase field approach in the lattice Boltzmann method for multiphase flow, a diffusive interface of approximately three lattice spacings between the fluid phases exists. If one drains a porous medium initially filled with wetting phase, one is left with a very thin layer of wetting

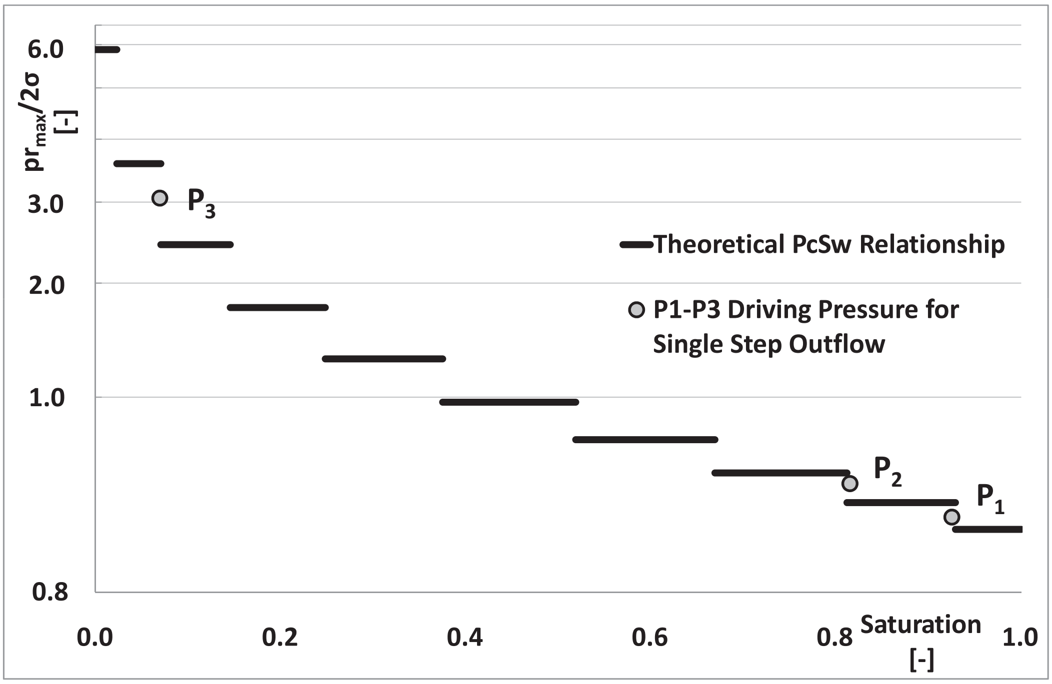


Figure 4.9.1 – Three different pressure loads for tube bundle (single step outflow)

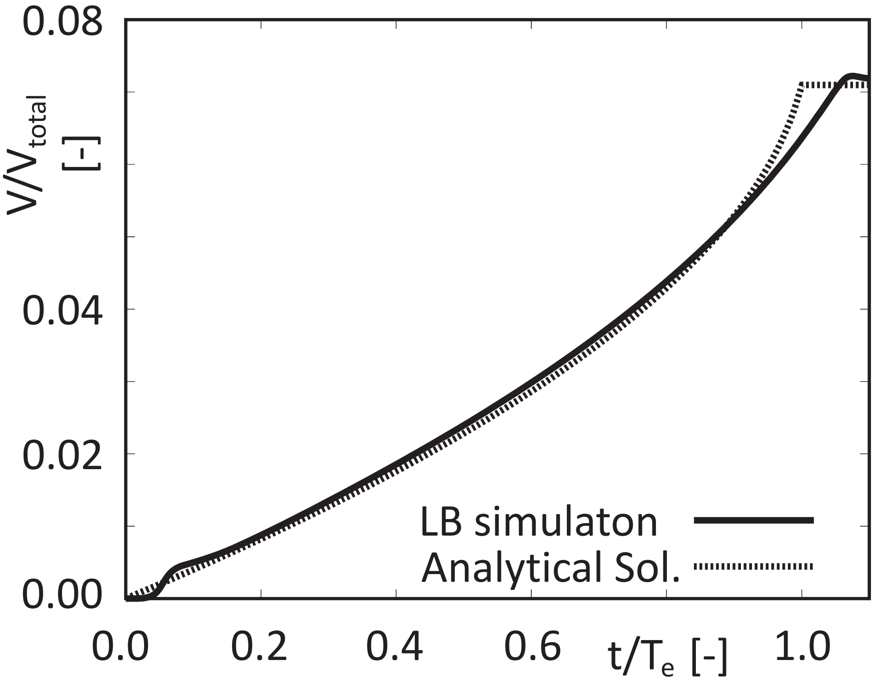


Figure 4.9.2 – Outflow curve for the multi step outflow experiment at pressure step P1

phase covering the matrix, meaning that the value of the order parameter ϕ (Equation 2.2.1) in drained cells next to the walls are close to the value of the non-wetting phase, but never drains

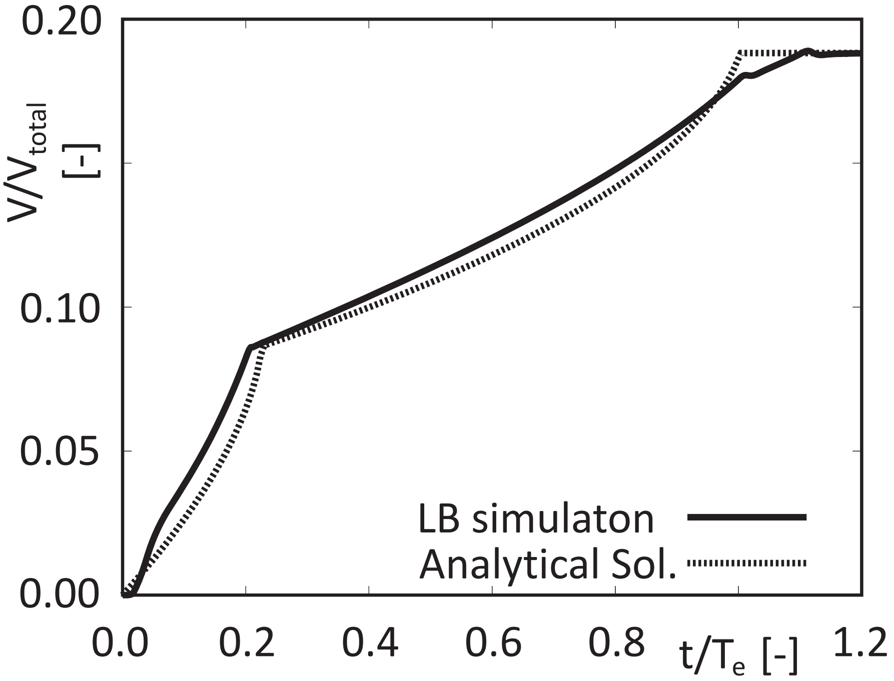


Figure 4.9.3 – Outflow curve for the multi step outflow experiment at pressure step P2

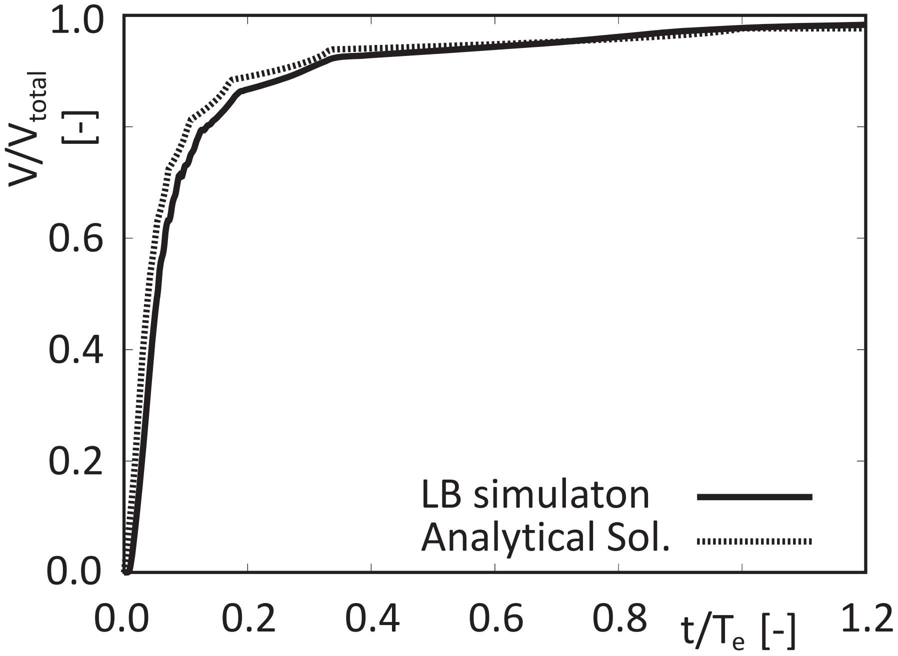


Figure 4.9.4 – Outflow curve for the multi step outflow experiment at pressure step P3

completely. Due to this effect one obtains a hydraulically conducting thin film[2] . Consequently, one can drain each geometrically connected porous medium almost completely - if a pressure is

---

[2] Film flow in this magnitude can only be avoided when using higher numerical grid resolutions, however, this would be a very demanding simulation setup. Even with grid refinement methods this would be a challenge, due to the extreme surface/volume ratio in porous media.

applied which is in the range of the maximal capillary pressure (Equation 4.10.1). For a given numerical grid, the smallest pore radius is approximated by the grid-spacing Δx, leading to a maximum pressure of:

$$p_{\Delta x} = \frac{2\,\sigma}{\Delta x}. \tag{4.10.1}$$

The behavior changes, if the grains in the porous medium are not connected anymore. If there are at least two lattice spacing between two geometric objects, film flow breaks and it is possible to obtain a residual water saturation. An approximate method to compute the irreducible wetting phase saturation is given in [128], page 205 *ff*, if the irreducible wetting phase exists in form of liquid bridges and the porous medium is composed of spheres. The value of the residual saturation S_r can be computed, by referring to [128], page 207, Table 4.2, and using linear interpolation. The related capillary pressure $p_{c,res}$ can be obtained by again referring to [128], page 128, Figure 4.12. Again, a body centered cubic array of spheres is used to test this behavior (Figure 4.10.1). The porosity ϕ can be computed from

$$\phi = 1 - \frac{8\,\pi\,r^3}{3\,L^3}, \tag{4.10.2}$$

and the closest distance a of two spheres from

$$\frac{a}{2\,r} = \frac{\sqrt{3}\,L}{4\,r} - 1. \tag{4.10.3}$$

Two different setups which are given in Table 4.10.1 have been tested. The material parameters have been $\mu_w = 1/200$ [Pa · s], $\mu_n = 1/500$ [Pa · s], $\sigma = 0.01$ [N m^{-1}] and the driving capillary pressure was increased linearly over time up to a maximum value of approximately $p_{max} = 18\sigma/(2\,r)$. The material parameters had the same values as already used for the simulations in real porous media performed in section 6.7. The domain was initially filled with a wetting phase and then drained by the increasing pressure. The grid resolution has been selected to 150^3 grid nodes, so that one sphere is resolved by approximately 40 grid points. In the first case the medium is drained almost completely, whereas in the second case a residual saturation of 0.13 is obtained. The distribution of the non wetting phase (NWP) is shown in Figure 4.10.2. The values of setup 2 are very close and indicate the validity of the approach. Nevertheless, a further investigation needs to be done but is beyond the scope of this work.

For a natural porous medium obtained by tomography methods it is very difficult to estimate the irreducible NWP saturation by lattice Boltzmann methods, since the medium is generally connected and the individual grains are not known. A method to identify single items in the structure is described in [81]. A possible future direction to estimate the irreducible NWP saturation by LB methods could use these methods and shrink the single items by two voxel layers in the numerical simulation grid. Finally, one has to perform several simulation runs on

Table 4.10.1 – Residual saturation, approximation by Schubert and the LB simulation

	$\frac{a}{2\,r}$	ϕ	$\frac{p_{c,res}\,2r}{\sigma}$, Schubert	S_r, Schubert	S_r, LB
Setup 1	0.0	0.32	≈ 9	0.06	0.0 (film flow)
Setup 2	0.083	0.46	≈ 4	0.12	0.13

successively refined grids, always keeping a distance of two voxel layers between the single items, until a convergent behavior is observed. Nevertheless, this is a very demanding simulation setup.

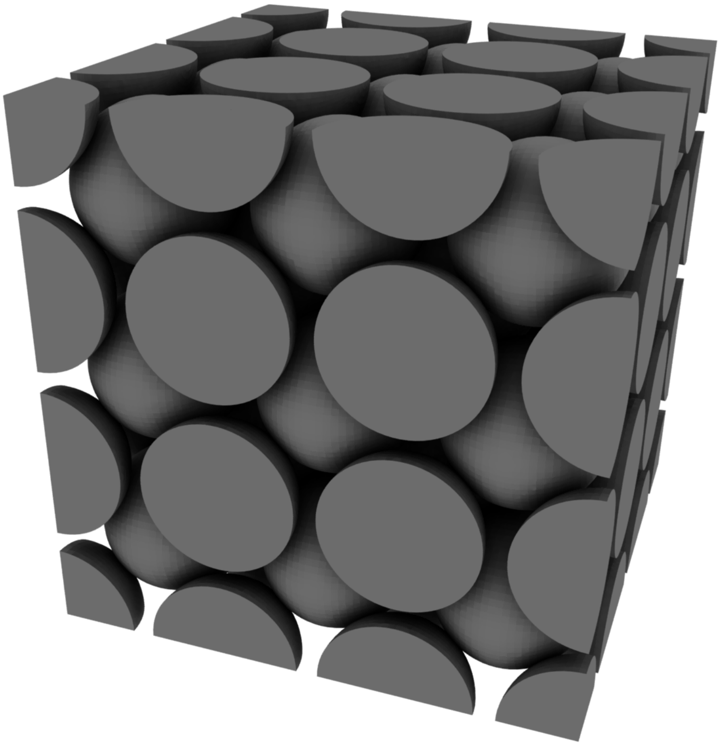


Figure 4.10.1 – Body Centered array of spheres, Left: geometric relations, Right: simulation domain

4.11 Attenuation of sound

The lattice Boltzmann method is also suitable for solving aero-acoustic problems, however the properties of the method regarding acoustic waves has only been investigated by a limited number of authors. [134] simulated 3D sound generation in a flue organ pipe using a simple LB algorithm in parallel. [25] simulated the propagation of linear sound waves using different boundary conditions and later [24] simulated the formation of shock waves. Lallemand and Luo [91] studied the wave propagation properties of several lattice Boltzmann schemes mainly focusing on the Multiple-Relaxation-Time model which has been continued by the same authors in 2003 [93]. Further works regarding the acoustic properties where e.g. [70] who predicted acoustic streaming around a cylinder between two plates of finite length. And more recently [161] studied the sound generation focusing on aero acoustics applications or [38] who predicted the sound field from a radiating waveguide.

In order to validate the implemented lattice Boltzmann algorithm and to prepare the kernel for acoustic simulations a few simple experiments have been executed. In order to estimate

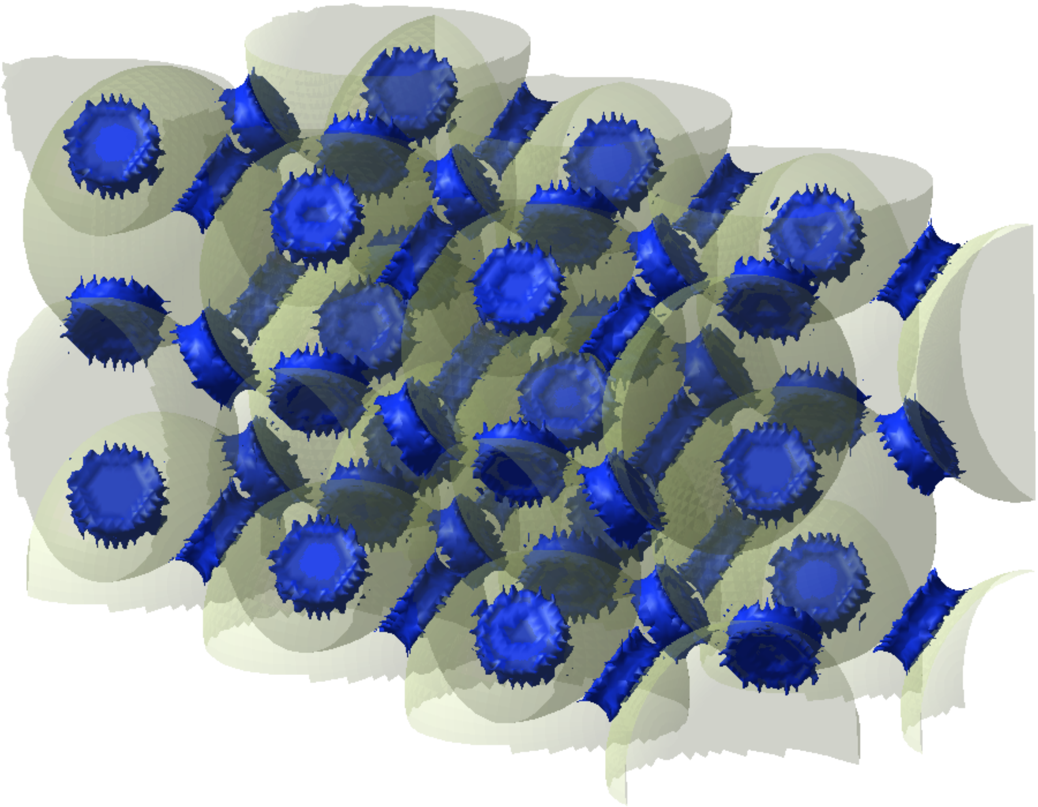


Figure 4.10.2 – Irreducible wetting phase saturation for setup 2

dispersion[3] and dissipation[4] , the attenuation of a sound wave in an unbound fluid has been computed. Periodic boundary conditions as described in section 3.4 have been applied to all domain boundaries imitating a completely unbound fluid. The pressure has been distributed in shape of a sinodial wave which has been initialized a priori into the bulk fluid. The wavelengths analyzed ranged from a few grid nodes to several hundreds. To compare the numerical results to an analytical solution, the following relation has been used [84]:

$$\alpha = \frac{2(2\pi)^3 E_w}{3} \tag{4.11.1}$$

where α is the spatial absorption coefficient and E_w^* is a dimensionless variable,

$$E_w^* = \frac{\eta}{c_s k \lambda^2} \tag{4.11.2}$$

and $\eta = \nu + \zeta/2$ where ν is the kinematic viscosity, ζ the bulk viscosity, c_s is the speed of sound, k the wave number, and λ the wavelength. The attenuation can be measured by simply evaluating the pressure difference. The comparison between the measured result and the analytical

---

[3] If a wave travels through a medium, dispersion is basically the dependency of the phase speed u_{ph} from the wavelength λ. Therefore, if the wave length becomes very short in terms of grid resolution (e.g. $\lambda << 5$ grid nodes) the numerical error increases rapidly.

[4] Dissipation means the transformation of one energy form to another. In acoustics typically the thermal losses of a sonic wave.

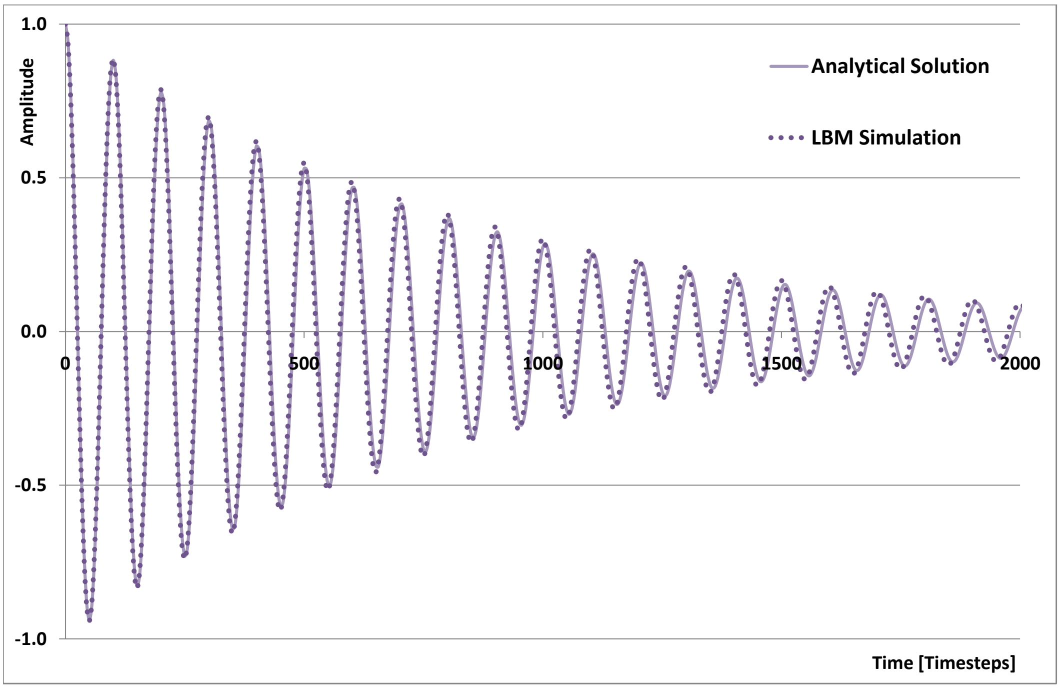


Figure 4.11.1 – Damping of a simulated sound wave in a pipe compared to the analytical prediction

prediction for one experiment is shown in Figure 4.11.1, exhibiting a very good agreement. All results taken during the experiments with a 3D lattice Boltzmann model are turned out to be consistent with 2D simulations executed by [25].

Furthermore the attenuation of a sound wave at rigid walls has been computed using an open rectangular domain with plane walls at top and bottom. The length of the domain has been dynamically allocated so that the wavelength of the initialized wave corresponds to the domain length. The next validation experiment introduces fluid/solid interaction boundary condition of no-slip type, meaning walls are added to the channel. That means that the fluid motion very close to a solid boundary is controlled by viscous stresses in a thin boundary layer across which there is a rapid adjustment in velocity to satisfy the no-slip condition. The viscous penetration depth or acoustic boundary-layer thickness, δ is expressed as:

$$\delta = \sqrt{2\eta/\rho_0\omega} \tag{4.11.3}$$

where ρ_0 is the equilibrium density. If the boundary layer thickness is much smaller than the pipe diameter, $\delta/\lambda \ll 1$, the damping rate per period caused by the confinement of the pipe is $\alpha = \pi(2E_p^*\omega')^{1/2}$, where E_p^* is

$$E_p^* = \frac{\rho\nu}{c_s k y_1^2}. \tag{4.11.4}$$

Combining this with the internal dissipation from equation 4.11.1, the total damping rate per period is:

$$\alpha = \pi\sqrt{2E_p^*\omega'} + (2\pi)^3 E_p^* F^2 \tag{4.11.5}$$

where F is the ratio of the pipe width to the wavelength of the sound, $F = y_1/\lambda$. The boundary condition has been implemented by using pressure boundaries at the in- and outflow, where the inflow condition is pulsating in a sinusoidal manner, inserting a sound wave into the fluid. The results also compare very good with the analytical prediction.

The experiment described above have also been conducted using a non uniform LB approach [49]. The channel has been setup with a refined grid in order to resolve the acoustic boundary layer more efficiently.

The results are of comparable accuracy, however, a convergence study could be executed, showing that a finer resolution of the acoustic boundary layer yields better results while the bulk fluid is still represented by a coarse grid. The influence of the boundary layer as well as the increasing accuracy of the measured results, showing a quadratic convergence behavior. While with the non uniform approach all calculations could be executed on a single workstation, the comparison at the very same resolution using an uniform grid, needed at least the utilization of 64 CPUs.

5 Grid Generation

Grid generation is the process of determining the coordinate transformation that maps the body-fitted non-uniform non-orthogonal physical space x, y, z, t into the transformed uniform orthogonal computational space, ξ, ζ, η, τ. It is often considered as the most important and most time consuming part of CFD simulation. The quality of the grid plays a direct role with respect to the quality of the analysis, regardless of the flow solver used.

In preparation of a fluid simulation the flow domain will be decomposed in typically small volume elements, sometimes referred to as cells or voxels. This process is also called discretization and the quality of a discretization is, beside the chosen numerical model, decisive for the accuracy of the computed results. Therefore, the discretization and the corresponding resolution of the numerical grid should be in a harmonic proportion to the physical state of a system. Additionally, the solver will be more robust and efficient when using a well constructed mesh. If simulations of fluid flow in natural porous media are performed, independently of the applied method, the geometry has to be prepared for the flow solver. In most cases a binarized voxel[1] matrix from a CT- or X-ray scanner is used and directly mapped to a numerical grid. Those grids typically have the same resolution as the original voxel matrix and therefore the solver computes all types of flow problems at a certain accuracy and efficiency. Also - at least if the voxel matrix consists of black and white values - the geometrical representation is reduced to first order, which is a serious loss in accuracy primarily if the initial material consisted of grey values. However, for a wide range of setups the full resolution of a sample is not necessarily needed or in the contrary it may be even useful to increase the numerical accuracy by refining the calculation grid. Typically, if an experiment aims e.g. on the determination of the saturated conductivity in a porous medium neither the whole sample nor the full resolution may be necessary.

5.1 Grid types

The numerical solution of partial differential equations requires some discretization of the field into a collection of points or elemental volumes (cells). In the lattice Boltzmann method mass fractions collide and propagate between adjacent cells and therefore only space filling grid types on a regular grid or - as the name already suggests - a lattice can be used. The most simple in 3D is a cube and a more most complex one the rhombic dodecahedron (see also chapter 2), the unit cell for the D3Q13 model. Besides the constrains of a space filling lattice, further demands are easy parallelization and low memory consumption. Therefore the choice of an efficient grid type is limited. Most widely used with LBM are probably uniform grids which are easy to implement and to parallelize. However, depending on the complexity of the system, uniform grids may not be sufficient for solving certain types of CFD problems. Especially if boundary layers issues appear, whether they are related to turbulence, film flows, thermal-, or acoustic problems. Also if various length scales should be simulated at once, uniform grids may not be sufficient. In that case the use of non-uniform meshes and adaptive approaches are required. For

---

[1] a voxel is an acronym derived from **Vo**lumetric Pi**xel**. The shape of a voxel is in most cases cubic.

lattice Boltzmann models the following grid types are suitable: block structured grids [45, 165] and unstructured tree-type grids [35, 36] with a priori refinement as well as adaptive approaches [50, 146]. The most promising approach is the use of block structured grids which combine the advantages of pure hierarchical and Cartesian grids (in this context they are also called hybrid grids). A sample application is shown in Figure 5.1.1 While this is only a very brief overview, a far more detailed description on drawbacks and benefits, implementation issues, and performance comparisons can be found in [49, 54].

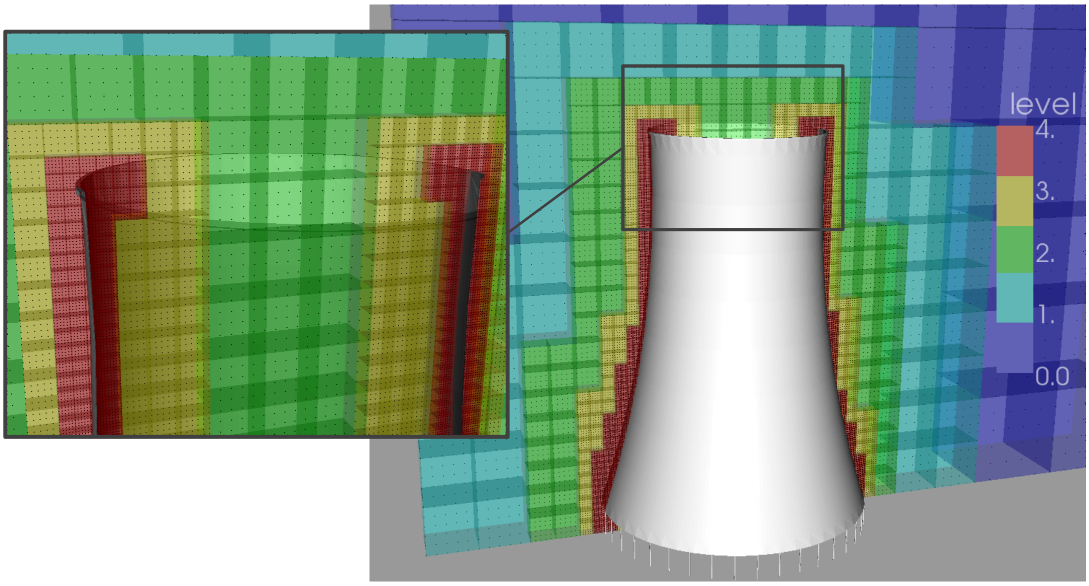


Figure 5.1.1 – refined block structured grid around a cooling tower

While grid refinement techniques are very useful in systems with a high ratio of bulk fluid vs. boundary layer regions, it may not be appropriate for simulations in porous media depending on the level of porosity. Therefore the focus in the grid generation process aims to a fast generation of robust and very large grids. Furthermore, and assuming an REV [2] is identified, boundary conditions of higher order are applied for convergence studies 6.4 in this thesis and in [5] have shown that numerical grid resolutions of 0.5 of the original domain may be sufficient to determine accurate results. Even uniform grids benefits from the low resolution technique.

The following section describes the grid generation process with focus of gaining computational efficiency by decoupling the numerical grid from a given voxel matrix and utilizing subgrid distances to take advantage of linear interpolated bounce back boundary conditions (see section 8.1).

---

[2] From Wikipedia, the free encyclopedia:

‘In hydrogeology, the representative elementary volume (REV) is the smallest volume over which a measurement can be made that will yield a value representative of the whole. Below REV, the parameter is not defined and the material can not be treated as a continuum. In order to establish a given porous medium’s properties, we are going to have to measure samples of the porous medium. If the sample is too small, the readings tend to oscillate. As we increase the sample size, the oscillations begin to dampen out. Eventually the sample size will become large enough that we begin to get consistent readings. This sample size is referred to as the representative elementary volume. If we continue to increase our sample size, measurement will remain stable until the sample size gets large enough that we begin to include other hydrostratigraphic layers. This is referred to as the maximum elementary volume (MEV).’

5.2 Surface reconstruction

During the actual grid generation process, the input model is usually either a facet mesh or a voxel matrix. Facet meshes are typically used if a geometry is artificial, e.g. it originates from a model which has been composed in a CAD/CAE[3] environment like cars, airplanes or parts of engines, etc. To map natural flow processes into a computer based model one needs geometric input data, which, in contrast to artificial structures, are derived [94] or even better, are directly digitized from natural structures. Such data can be obtained by using tomography scanners which are capable of transforming details of natural structures into digital data. Such structures are typically porous materials like soils, sands, asphalt or even bones or wood. If flow simulations are to be performed in these materials they are typically executed directly on the voxel matrix. This has the disadvantage that the established grid, and its resolution and discretization cannot be changed. Therefore the accuracy, the speed, and the computational effort depend not only on the type of model, but also on the scan resolution. In this work a more elaborated approach has been developed, were the input geometry is transformed into a parameterized surface and the computational grid can therefore be decoupled from the original voxel matrix [5].
As already mentioned above, the triangle mesh used as input data may be created artificially in a modeling tool or is obtained through surface reconstruction from a digitized data set. In the second case the data is commonly organized in 3D matrices or in 2D stacked picture slices. Depending on the scanning procedure and the applied image processing the voxel data consists of black and white (b/w) or grey values. Such sets of data representing volumes are very memory demanding and sometimes very difficult to process. Typical resolutions of CT scans ranges between hundreds, or even thousands of voxels in each dimension and the size of the samples ranges between a few millimeters up to the size of a human body. Therefore a sample of sand which has been scanned with a resolution of 10 microns and an edge length of 10 mm results in a data set of 1000^3 voxel. This matrix consumes a memory size of $1000 \times 1000 \times 1000 \times 4$ Byte ≈ 4 GByte if grey values are used. If this matrix is triangulated - depending on the geometric complexity and the algorithms used - the resulting mesh size ranges from a few million up to many hundred million triangles. A non-compressed, non binary file containing a parameterized surface can easily have sizes of 10 to 20 GByte. For the triangulation of a voxel matrix or any arbitrary scalar field various algorithms can be utilized, however, in the majority they are all following the same principle. A well known algorithm is based on the Marching cubes concept (Figure 5.2.1), which is considered the reference technique for reconstructing surfaces from scalar fields. The algorithm has been introduced by [99] as a method for visualizing data from 3D matrices. This method has been further developed by [33] and [97] to resolve topological disambiguities. As an alternative the class of marching tetrahedras[4] can be proposed. The Marching Cube algorithm classifies the points in the matrix as either positive or negative depending on a threshold value,

---

[3] Computer-aided design (CAD) is the use of computer technology to aid in the design and particularly the drafting (technical drawing and engineering drawing) of a part or product, including entire buildings. It is both a visual (or drawing) and symbol-based method of communication whose conventions are particular to a specific technical field.

[4] In: Efficient implementation of Marching Cubes' cases with topological guarantees, by T. Lewiner et al.: 'Another range of techniques for isosurface generation is based on tetrahedra [151], as opposed to cubes. Those methods guarantee the topological consistency, and have a small lookup table. However, they have many drawbacks. They generate much more triangles, with a weaker geometrical accuracy of the result: the cubes tiling is segmented even in obvious configurations, and the vertex position cannot be adjusted to fit the geometrical tri-linear approximation as we do with cubes. Moreover, the ambiguity resolution which is hidden in those methods leads to slower algorithms, which are more difficult to speed up with hardware implementation.'

the so called iso-value. 8 cell centered voxels are then combined to a cube which is then processed in the next step using a lookup table for tiling the surface. The 8 corners of every cell result in $2^8 = 256$ possible different configurations, however due to reflections and symmetrical rotations 15 unique cases remain. Figure 5.2.1 shows the possible unique configurations of the marching cube algorithm. This method has been improved and optimized in many concerns, mainly to reduce the amount of cube cells that have to be inspected, the amount of triangles which are created and to increase the consistency of the resulting mesh [112, 127]. However, most methods are using a very simple lookup table which does not guarantee topological consistency, meaning, that certain ambiguous configurations of the scalar field and of the inspected cube cell respectively can lead to unexpected results (Figure 5.2.3). These effects can be holes in the triangulated surface, disoriented or duplicated triangles which can subsequently disturb the grid generation process. While duplicates of triangles have almost no impact, surface holes as well as disoriented triangles can severely affect the grid generation process.

Natural porous media are characterized by an integral surface, because no grain can exist isolated and it is always connected to at least one other grain. Therefore, if the surface is perforated (even only in one place), it is very likely that the recursive filling algorithm will fail, because structure and pore space can not be identified unambiguously. A disoriented triangle, meaning a triangle whose surface normal points to the inside instead to the outside which can lead to another ambiguity, namely that the closest grid cell is marked as fluid/wall interface node where it is in fact a solid/wall node. Furthermore, non initialized grid nodes, which have not been reached by the first fill attempt (sometimes boundary nodes prevent the fill algorithm from reaching isolated pore space or small portions of void space inside a grain) are filled with the wrong cell type. To circumvent these issues, the grid generation process is based on a Marching Cube algorithm with topological guarantees [97]. To avoid cracks in the isosurface, this implementation tests for face ambiguities (Figure 5.2.2) in a cube and if one detected the adjacent cubes will be analyzed to guarantee a coherent transition of the generated

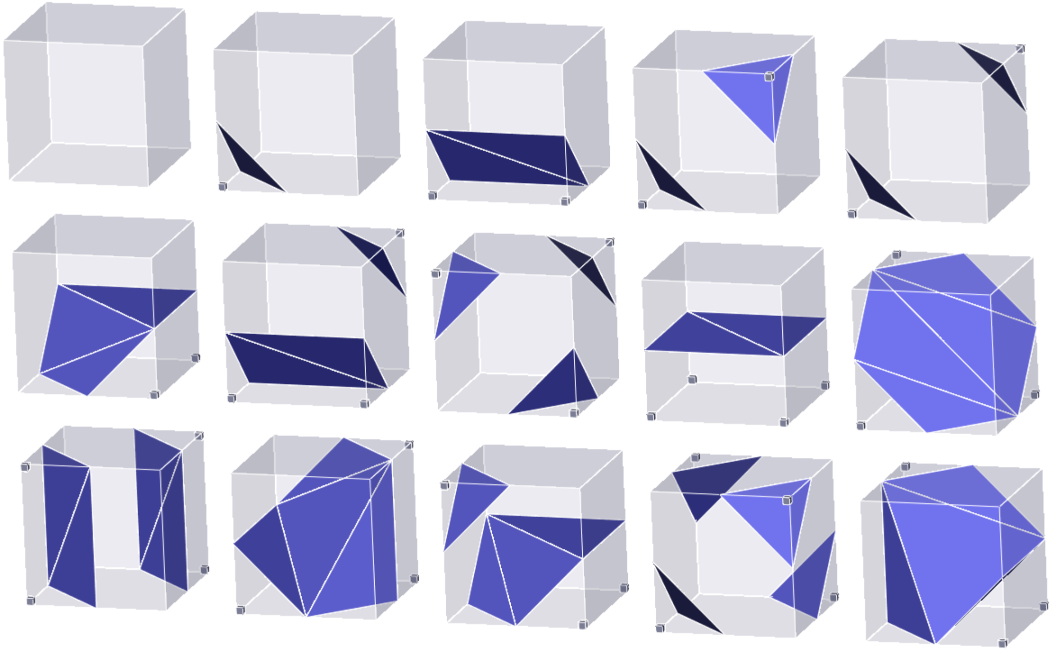


Figure 5.2.1 – Marching Cube base configurations

surface. This process in combination with an optimized lookup table result in an algorithm which produces a more consistent mesh needed especially for the numerical grid generation in a porous medium. However, even though this proposed algorithm produces a high quality mesh, it is not completely free of inconsistencies. Digitized samples of porous media originating from CT-scans often contain scanning artifacts besides their inherent highly complex structure. Therefore, holes and disoriented triangles can appear in the reconstructed surface. In order to avoid side effect during the grid generation process some additional algorithms which can detect certain inconsistencies especially suited for the reconstruction of natural porous media surfaces have been developed and implemented into the grid generation procedure. Some of them are algorithms for:

- find and close holes in the reconstructed surface by adding gap filling triangles
- find and correct triangle edges which are not connected to the mesh
- detect grid nodes which can not be unambiguously connected to fluid or solid space due to rounding errors during the subgrid distance computation between node and triangle surface

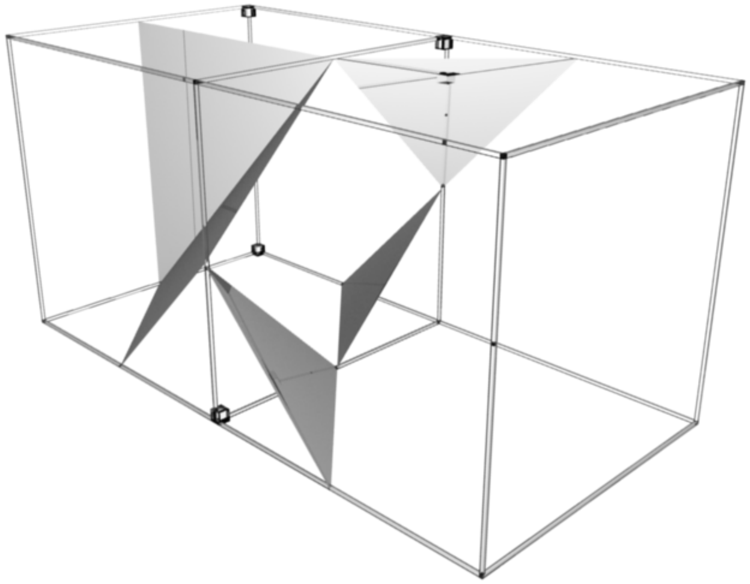


Figure 5.2.2 – Crack in surface due to ambiguous configuration

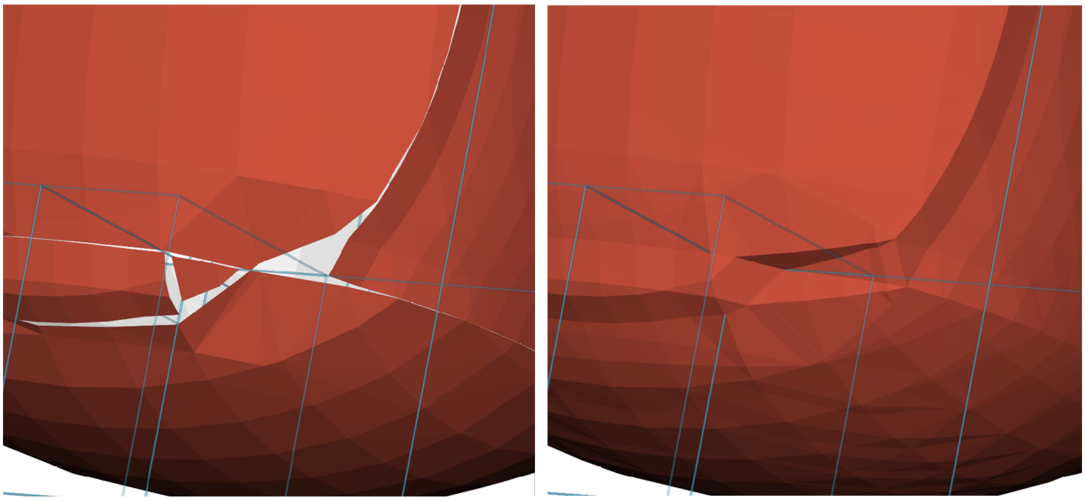


Figure 5.2.3 – Cracks in an isosurface due to disambiguities in the underlying voxel matrix

Marching Cubes Patent Issues:
This algorithm was the prime example in the graphics field of the woes of patenting software, patented despite being a relatively obvious solution to the surface-generation problem. Another similar algorithm was developed, called marching tetrahedrons, in order to circumvent the patent as well as a minor ambiguity problem of marching cubes with some cube configurations. This patent has recently expired, and it is legal for the graphics community to use it now without royalties since more than 20 years have passed from its filing date (June 5, 1985).
Marching Cubes, US Patent Office entry

http://patft.uspto.gov/netacgi/nph-Parser?patentnumber=4710876

(source: `http://en.wikipedia.org/wiki/Marching_cubes`)

5.3 Subgrid distances

Nowadays scans of natural porous media are delivered as datasets which consists of grey values representing the volume of a sample. Fortunately, stair case patterns as a geometric artifact, belongs to the past. However, if a voxel is mapped to a numerical grid in a lattice Boltzmann context, the grid cell has only a binary type; fluid or solid, the distance information is lost and the geometric accuracy is limited to first order. To circumvent these constrains solid wall boundary conditions of higher order can be applied, assuming the distance information can be preserved. While the grid generation process depends on reconstructed or parametrized surfaces the intersection points between the cell stencil and the surface can be computed (Figure 5.3.2) and stored in a list based data structure .

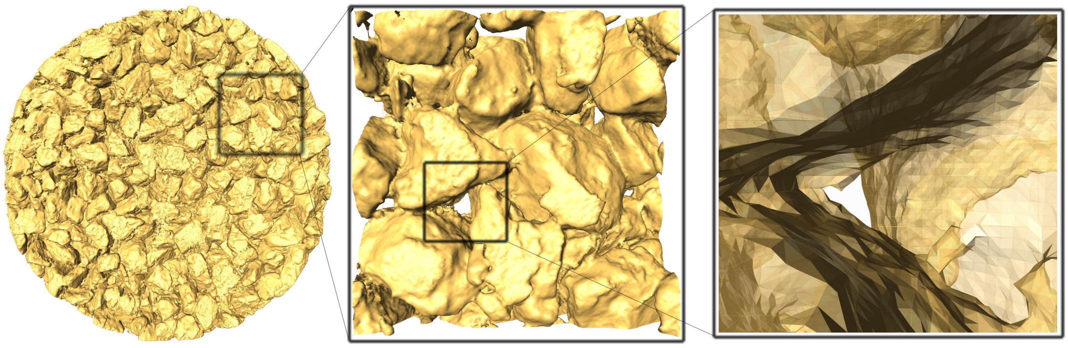


Figure 5.3.1 – Triangle mesh generated from a sample of porous asphalt with $100mm$ diameter

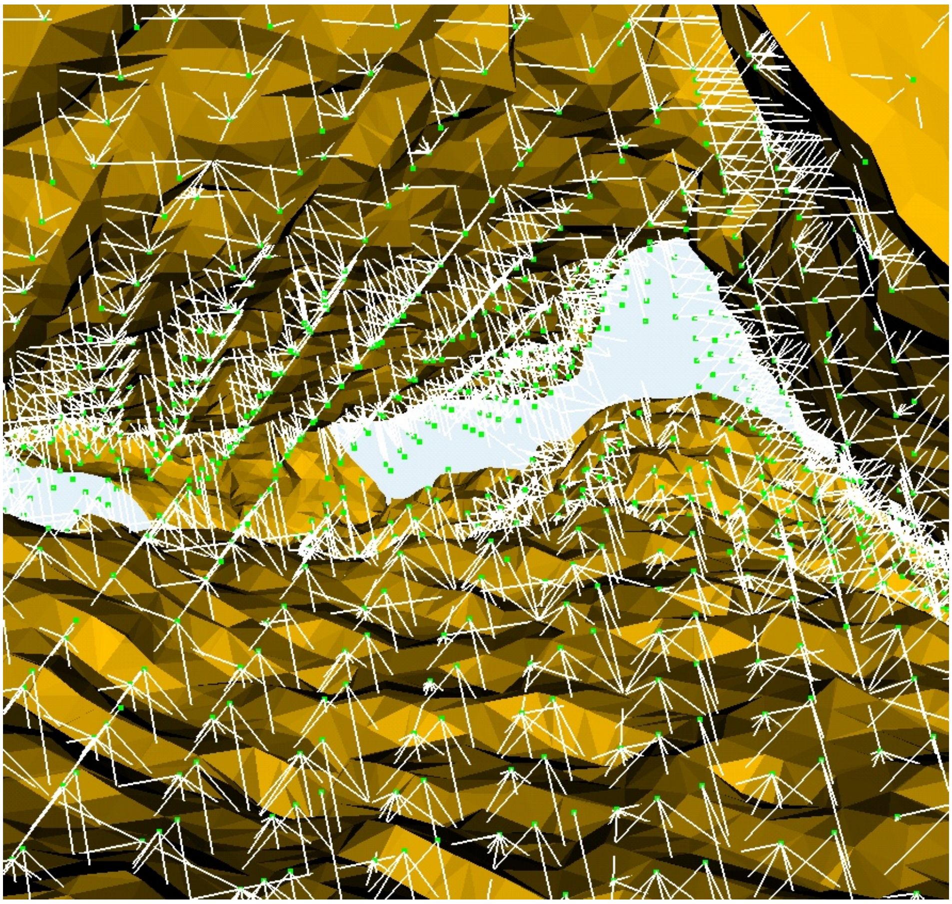


Figure 5.3.2 – Subsection of triangle mesh of a porous media showing boundary nodes and associated sub-grid distances

5.4 Normal generation

The surface normals of a triangle mesh are needed for multiple reasons. Typically the normal information is used for lighting in terms of graphics representation. If a triangle mesh is displayed, the visualization process needs information, such as the vertex normal to compute lighting, Gouraud shading, and texturing effects. For a plain polygon (such as a triangle), a surface normal can be calculated as the vector cross product of two (non-parallel) edges of the polygon. The vector direction is determined by the order in which the vertices are defined and by whether the coordinate system is right- or left-handed. The face normal points away from the front side of the face. This determines the in- and outside of a mesh, respectively solid and fluid or structure and pore. This is one crucial reason why normals are needed for the grid generation process. Even though the calculation of the normals is easy; it is far more difficult to assure a consistent mesh for highly complex geometries such as reconstructed porous media originating from tomography scans. The reconstructed surface of this kind of geometries is often compromised by holes (Figure 5.2.3 or triangles which are oriented in the wrong direction, meaning there vertices are in a random order (Figure 5.4.1). For a smooth and consistent grid generation process these abnormal triangles have to be identified and, if possible, modified. To make the triangle meshes and their derived numerical grids consistent some effort has been made. Most parts of the grid generation code are functions to either maintain the consistency or to reconstruct the integrity. Besides visualization and the determination of pore space and structure, the normal information is also needed if slip boundary conditions are applied. In that case the normal corresponding to a boundary node is additionally written to a file which contains also the subgrid distance information which can then be processed by the simulation kernel.

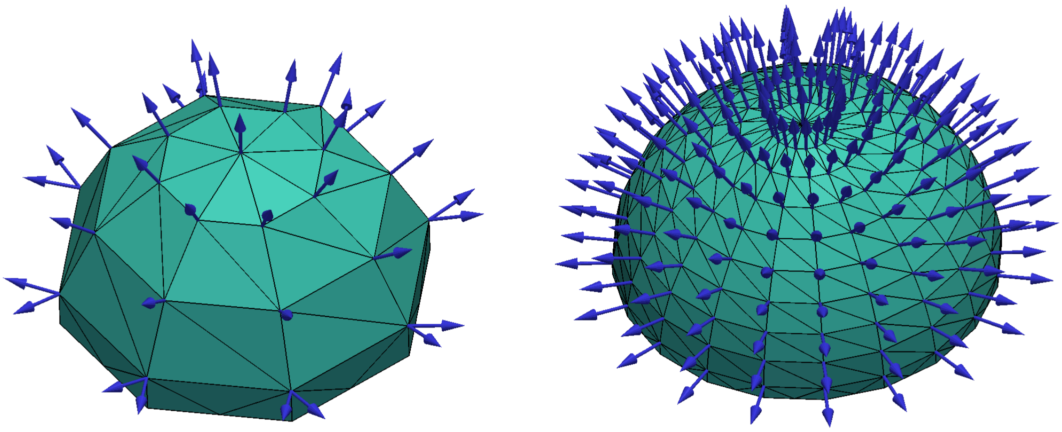

(c) Vertex normals (d) Disordered normal configuration

Figure 5.4.1 – Vertex and surface normals of polygonal mesh

6 Flow in porous media

The lattice Boltzmann method is particularly useful for problems of single and multiphase flows in porous media because of its capability of handling complex geometrical boundaries and varying physical parameters, including viscosity, surface tension and wettability. These distinct abilities promote the LBM to a state-of-the-art procedure for solving transient flow problems in porous media. This chapter describes numerical single- and multiphase experiments performed in porous media to determine specific material parameters and constitutive relationships, such as saturated permeabilities, capillary pressure-saturation relationships, hysteretic effects during drainage and imbibition (section 6.3 and 6.7).

6.1 Classic experiments and alternative simulation methods

The classical approach to determine hydraulic properties of porous media is to carry out multistep inflow/outflow experiments, where a porous medium sample is saturated with water and afterwards drained by decreasing the boundary pressure stepwise. Inverse models are then applied to determine the hydraulic properties from the measured outflow curve (i.e. [115]). Such types of experiments and a typical outflow curve is illustrated in Figure 6.1.1 and 6.1.2 and can also be found in [7]. From those measurements one can obtain the absolute and relative permeabilities, a hysteresis curve of the drainage and imbibition process as well as data related to porosity. Usually, for single phase measurements, water (or sometimes alcohol) is used to saturate a sample. However, the wetting of a dry sample to 100% saturation is a challenge. Alternativly one can measure e.g. the hydraulic conductivity by using a gaseous phase such as air, nitrogen, or CO_2. Due to the changed slip properties of gas at a solid wall one has to correct slip flow errors also known as the Klinkenberg[1] effect. Gas as a carrier for experimental measurements is frequently used in petroleum engineering as well as in geophysics and primarily in porous material with a very low intrinsic permeability.

Although the lattice Boltzmann Method is very well suited for numerical experiments in complex geometries such as porous material at the pore scale, there are of course alternatives for the solutions of Stokes flow problems [108]. Besides Finite Element- and Finite Difference-Methods [32] there is the class of Pore Network Models [74, 96].

A fast way to determine various constitutive relationships at the pore scale is the use of pore network models or full morphological pore network models (MPNM). These models need considerably less computational effort than e.g. lattice Boltzmann modeling. However, they are

---

[1] In 1941 Klinkenberg [86] discovered that gas-permeability as is relatively higher than fluid-permeability, and he interpreted this phenomena as "slip flow" between gas molecules and solid walls. He stated that gas molecules collide each other and to pore-walls during traveling through the pore medium. Therefore this additional flux due to the gas flow at the wall surface, which is called "slip flow", causes an increased flow rate. Another factor is the moisture content of the material, which can obviously affect the gas permeability. Comparisons of gas and water permeability measurements have been made in [140].

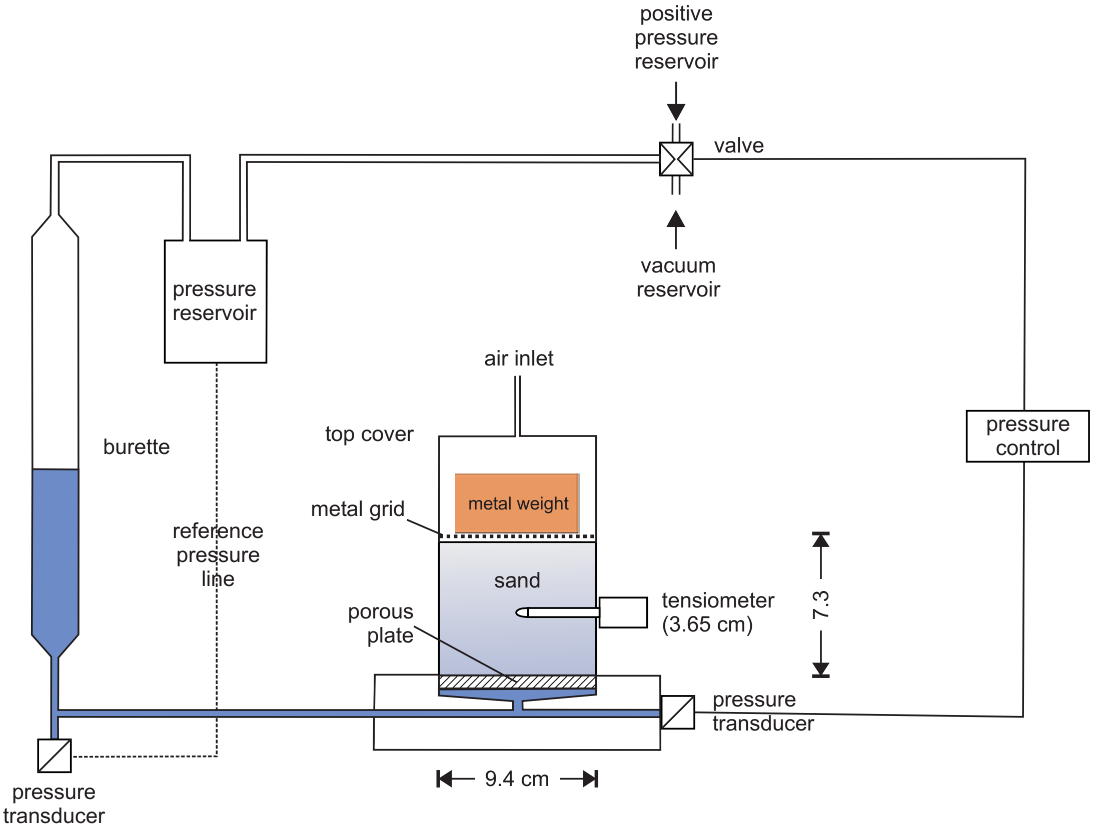


Figure 6.1.1 – Experimental setup of the multi-step outflow/inflow experiments [7].

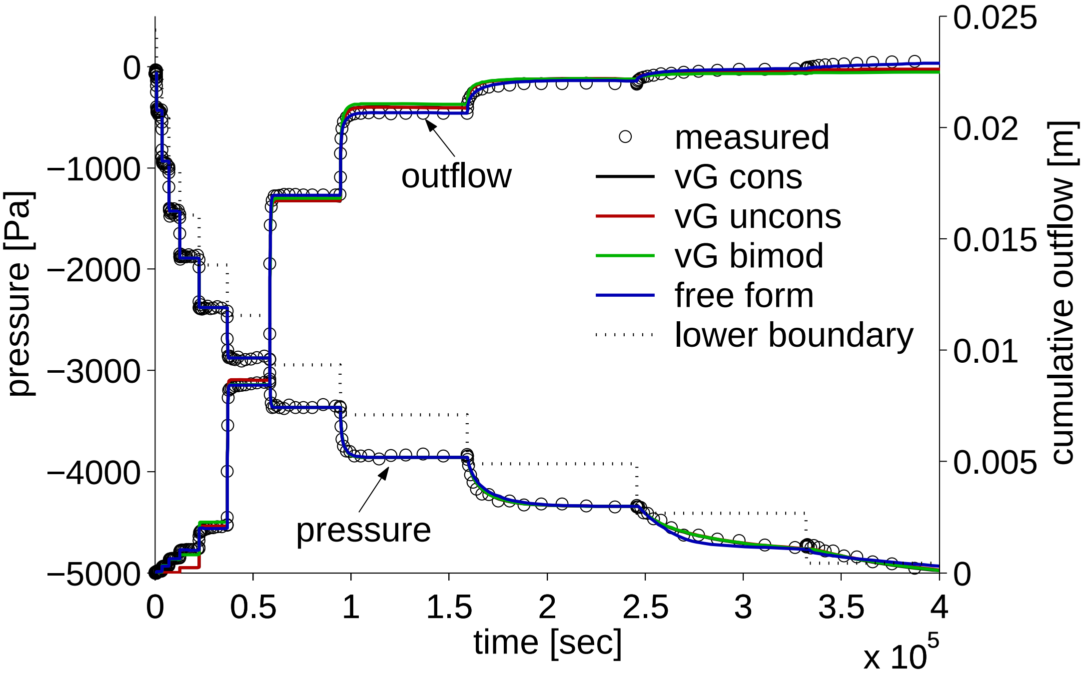


Figure 6.1.2 – Typical outflow curve of a multi-step outflow experiment [7].

unable to simulate dynamic flow effects and the choice of the inserted structural element is difficult. Also, depending of the used element, a large modeling error is introduced. Nevertheless, MPN models are excellently suited for comparison to classical experiments as well as to LB simulations [7, 157].

The Morphological Pore Network Model - Basic Concepts

Because a Morphological Pore Network Model is used to compare results from experiment and simulation a short description of the basics is given here: *The computation of the water retention curve with a morphological pore network model is based on a two-step procedure. Firstly, a size is assigned to each voxel of the pore space. In a second step, the replacement of a fluid phase by an invading fluid is described as an invasion percolation process. To determine the pore size, structural elements are inserted into the pore space. In case of the drainage step that is controlled by the narrowest structures, spheres with an increasing diameter are centered at each voxel of the pore space until the sphere is touching the solid phase. The radius of the sphere corresponds to the distance to the solid phase. A voxel close to the solid is the center of a small sphere but may be an element of a larger structural element as well. In that case, the larger value is assigned to the pore voxel. The diameter of the structural element was increased in increments of two voxel sizes. For that purpose, the pore size is a multiple of the voxel size. In case of the wetting process, other structural elements has been tested as well. A wetting process is affected by the widest element. For that purpose, two dimensional disks as structural elements has been used. They were inserted into planes orthogonal to the three main directions.*

For more details on Morphological Pore Network Models see [74, 96].

6.2 Single phase simulations

Many natural substances like rocks, soils, biological tissues (e.g. bones), and man made materials such as cements, foams and ceramics or even paper can be considered as porous matrix material. Experimental measurements or simulations in a single phase context aim mostly on the determination of hydraulic conductivity or permeability. The following list shows some examples where knowledge about the transmissibility of porous material is essential:

- The hydraulic conductivity is one of the most important parameters for studying the barrier's behavior in waste disposal facilities.
- The permeability of package materials needs to be matched with the sensitivity of the package contents and the specified durability. Some packages must have nearly hermetic seals while others can (and sometimes must) be selectively permeable.
- The permeability of rock is rather important for the petroleum industry, because oil production properties strongly depend on it.
- The transport of contaminants in soil is an important subject in environmental sciences and therefore the permeability properties.

These small selection of examples demonstrate why knowledge about the exact permeation rates is essential. The determination of permeability of porous material are a subject of most common interest and therefore have emerged as a separate field of study. The next section describes the simulation of single phase flow in natural porous media using lattice Boltzmann methods and the corresponding results.

6.3 Saturated permeability of a porous media

With the grid generation procedure described in chapter 5 and the resulting subgrid distances between the nodes of the Cartesian grid and the planar triangle surfaces one can efficiently compute permeabilities with a second order accurate lattice Boltzmann flow solver. Because the permeability depends strongly on the porosity of a porous medium, the dependence between the iso-level threshold used by the marching cube algorithm [97] to construct the triangulated surface and the resulting pore volume has been investigated. The permeability results may also depend on the chosen section within the whole sample of size 800^3 voxels. To estimate the sensitivity of the results on the chosen section, two subcubes with extremal properties were selected and numerical permeability measurements have been performed. For that purpose, the characteristic geometrical properties for various subcubes have been computed. The origin of the analyzed subcube was shifted in intervals of 20 voxels in x-, y- and z-direction from position $0, 0, 0$ to $600, 600, 600$ within the large image matrix. Totally, 31^3=29791 subcubes were investigated. The choice of the geometrical properties were based on the following hypothesis:

- the fluid distribution depends on the pore volume that is quantified as porosity ϕ and
- on the pore size that is orthogonal to the pressure applied in z-direction that was quantified using the chord length distribution in x and y-direction.

A chord is a line entirely within the pore space between two solid particles. To determine the chord length distribution, the voxels that are arranged in a chord of length l are counted and divided by the total number of voxels of the pore space. As a characteristic length for the chord length distribution, the median (the median is described as the value separating the lower from the upper half of a sample/distribution) of the distributions C_x and C_y in x- and y-direction were calculated. While other geometrical properties could be used to characterize the pore size as well, the chord length is the most effective to compute. The product $\Gamma = \phi\ C_x\ C_y$ that has the dimension of an area has been calculated for all subcubes. The minimum and maximum value was $113.8\ r_v^2$ and $168.4\ r_v^2$ (originating from a voxel size r_v of 11 $[\mu m]$). The subcubes with a maxima and a minima of Γ are denoted as 'maximum' and 'minimum' geometry respectively and labeled as A_{Min} and A_{Max}. A subcube (also of size 200^3) from the center of the sample has the value $\Gamma = 134.0\, r_v^2$ and is therefore labeled as A_{Med} (A_{Min}, A_{Max} are shown in Figure 6.7.3). After having identified the extremal sections of the whole 800^3 voxel large sample the reference porosity of the different subcubes A_{Min}, A_{Max} and A_{Med} have been computed just by counting the fluid (value 0) and solid cells (value 1). The volume of the triangulated pore space was computed by integrating over the entire surface using the divergence theorem [48]. In Figure 6.3.1 the results are shown and it turned out that the default iso-level value of 0.5 (from an interval between 0 and 1) fits the reference pore volume of the voxel matrix best in all cases. After assuring that a random iso-level threshold does sensitively impact the outcome of the simulation, for all further experiments the iso-level has been fixed to 0.5.

In Table 6.3.1 the permeability computed for A_{Min}, A_{Max} and A_{Med} using different grid res-

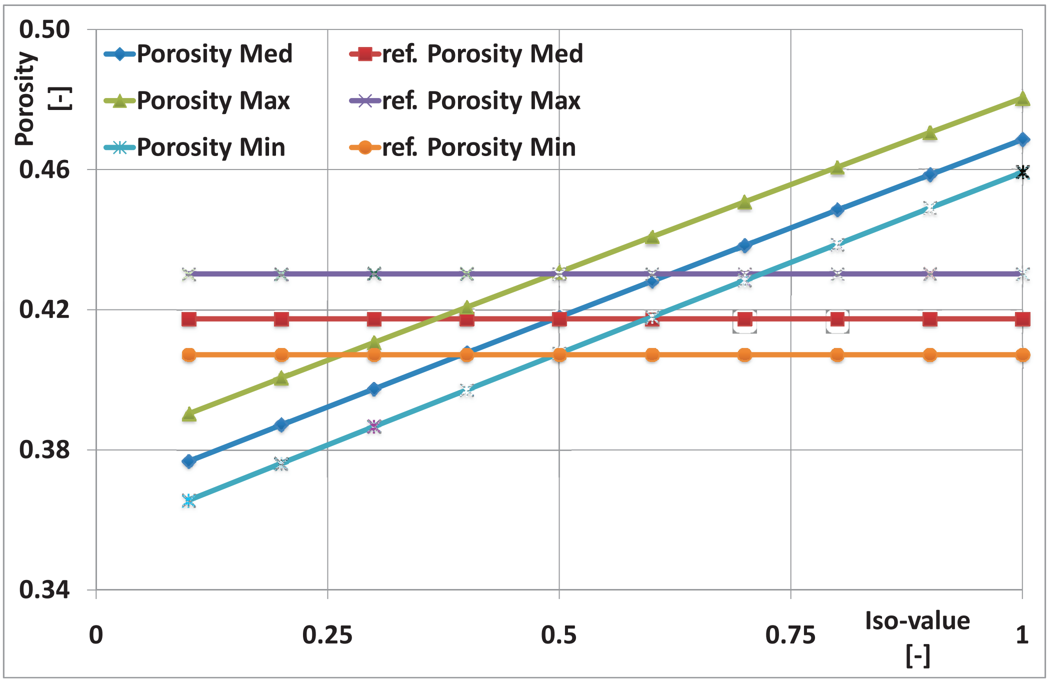


Figure 6.3.1 – Dependence of porosity on the iso-level threshold for three subcubes A_{Min}, A_{Max} and A_{Med}.

olutions are shown. An approximation for the permeability can be gained from the Kozeny equation [87]

$$k = \frac{r_{sv}^2 \phi}{5} \tag{6.3.1}$$

where r_{sv} is the ratio of the pore space volume to the wetted surface and ϕ is the porosity. Since we have a triangulated surface of the porous medium, we can easily compute the integral surface by using the divergence theorem.

The numerical results are very close to the values predicted by the Kozeny equation (Equation 6.3.1) and therefore indicate the high accuracy of the approximation. The values are not compared to the experimental results since the columns were not completely saturated[2] and therefore the deviation is too large.

---

[2] In geo- or soil-physical experiments it is very difficult to saturate a certain sample to 100%, since the displacing process of the non-wetting phase, such as air, depends strongly (among other effects) on the velocity of the intruding wetting phase (such as water). The impressed wetting phase outruns various pores still filled by non-wetting fluid and therefore almost every time such an experiment is conducted, non-wetting phase is surrounded by wetting phase leading to a not fully saturated porous medium. Permeability measurements in porous media which are not fully saturated are not comparable to fully saturated samples, because already very small inclusions of air alter the results significantly

Table 6.3.1 – Saturated permeability, LB simulation and Kozeny's equation, relative error with respect to the finest resolution of 399^3 grid nodes

Geometry	m	ϕ	k, Eq. 6.3.1	k, LB, 151^3	k, LB, 299^3	k, LB, 399^3	rel. Error
A_{Min}	0.388	3.54E-5	9.74E-11	9.96E-11	9.66E-11	9.56E-11	1.8%
A_{Med}	0.399	3.71E-5	1.10E-10	1.11E-10	1.07E-10	1.06E-10	3.6%
A_{Max}	0.412	3.93E-5	1.28E-10	1.35E-10	1.31E-10	1.30E-10	1.5%

6.4 Convergence study for a natural porous medium

Present neutron tomography[3] or X-rays[4] techniques are capable of scanning grey value images reaching voxel resolutions of a few microns. Thus one can perform fluid simulations in natural porous media with voxel resolutions up to 10^9 grid points. However, the demands of a simulation with a computational grid of the same size are very high. To demonstrate the increase in efficiency by decoupling the numerical grid from the geometrical voxel matrix a convergence study based on tomography scans of sand cubes has been performed. The data sources were subsets of the sand sample introduced in section 6.3 with a size of 250^3 which corresponds to a size of 1.25 mm^3 which is approximately 5-7 times the diameter of a typical pore diameter. Multiple simulations at different resolutions have been carried out and the corresponding results are shown in Table 6.4.1 and Figure 6.4.1. The resolution shown in the first column indicates the factor in grid size from the original voxel-based geometry resolution of 250. The reference permeability of the medium has been calculated using a Richardson extrapolation [22].

6.5 Multiphase simulations

While performing simulations within a multiphase context in natural porous material one is confronted with additional forces impacting the systems behavior, most important are the surface

---

[3] Neutron tomography:
Neutron tomography (NT) is a similar technique like X-ray tomography. The advantage is the strong interaction of neutrons with light materials and their excellent penetration properties of heavy materials. While X-rays are attenuated more effectively by heavier materials like metals, neutrons make it possible to image some light materials such as hydrogenous substances with high contrast. Therefore NT is especially suited for scanning partly saturated porous media. For a more detailed description of this technique see [123] and the references therein.

[4] X-Ray tomography:
X-Rays from a synchrotron source is radiation with an extremely narrow frequency band (monochromatic radiation). In contrast to conventional X-Ray sources with polychromatic radiation, it is possible to produce an image quantitatively indicating the distribution of a particular chemical element as well as images with a very fine resolution. These properties promote this method besides the relatively new method of neutron tomography to a state-of-the-art, non invasive imaging technique. Details and comparisons of methods based on different X-Ray sources, which have been applied to partly saturated porous media, can be found in [162]

Table 6.4.1 – Permeability results depending on different numerical grid resolutions and on boundary conditions.

Resolution	Voxel	rel. K (SBB)	rel. K (LIBB)	rel. error SBB [%]	rel. error LIBB [%]
1.75	436	3.8137092E-05	4.1587915E-05	8.915E+00	1.222E-01
1.5	374	3.7605696E-05	4.1800425E-05	1.045E+01	6.300E-01
1.0	250	3.5834207E-05	4.2637626E-05	1.591E+01	2.581E+00
0.75	187	3.4072388E-05	4.3666093E-05	2.191E+01	4.876E+00
0.5	125	2.9753718E-05	4.6620740E-05	3.960E+01	1.090E+01
0.3	76	1.7191929E-05	5.4151301E-05	1.416E+02	2.329E+01

tension σ and, if existent, gravity and therefore buoyancy. These forces and their mutual ratios are usually expressed in dimensionless numbers. For two phase flow in porous media on the microscale the important dimensionless parameters (in the absence of gravity) are the ratio of the dynamic viscosities or the ratio of mobility and the capillary number:

$$M = \mu_n/\mu_w, \quad Ca = u_w\mu_w/\sigma, \tag{6.5.1}$$

where u_w is the wetting phase Darcy velocity, μ_n and μ_w are the dynamic viscosities of the non-wetting and the wetting phase and σ is the interfacial tension. The contact angle is set to 0° in all simulations, but other values are possible with lattice Boltzmann methods [148]. The Bond Number Bo is of secondary importance, because the fluid flow in porous media is dominated by surface tension instead of gravity.

The fluid solver rescales the system in such a way, that numerical errors are reduced (avoiding very small and/or large numerical values). Since this rescaling is inherent to most LB implementations, the procedure for rescaling two systems labeled as 1 and 2 with equal dimensionless parameters is presented below. The pressures between the two systems 1 and 2 are then related by

$$\frac{p_1 r_1}{\sigma_1} = \frac{p_2 r_2}{\sigma_2}, \tag{6.5.2}$$

where $\{r_k, k = 1, 2\}$ is a reference length and the time scales are related by

$$\frac{T_1\,\sigma_1}{r_1\,\mu_1} = \frac{T_2\sigma_2}{r_2\,\mu_2}. \tag{6.5.3}$$

For air/water systems the ratio of $M \approx 1/50$ and the capillary number depends on the problem considered, but in most cases a low capillary number is desired. This is a challenge for numerical simulations, since there is a lower bound for the viscosity and an upper bound for the surface tension, where a stable simulation can be performed. Choosing a smaller velocity is possible, however, this leads also to smaller time steps to be calculated and therefore to more computational effort. But for some problems M is insignificant for the physical process and for many problems ratios of $M \cong 1/10$ are sufficient to reproduce the main physical effects. For

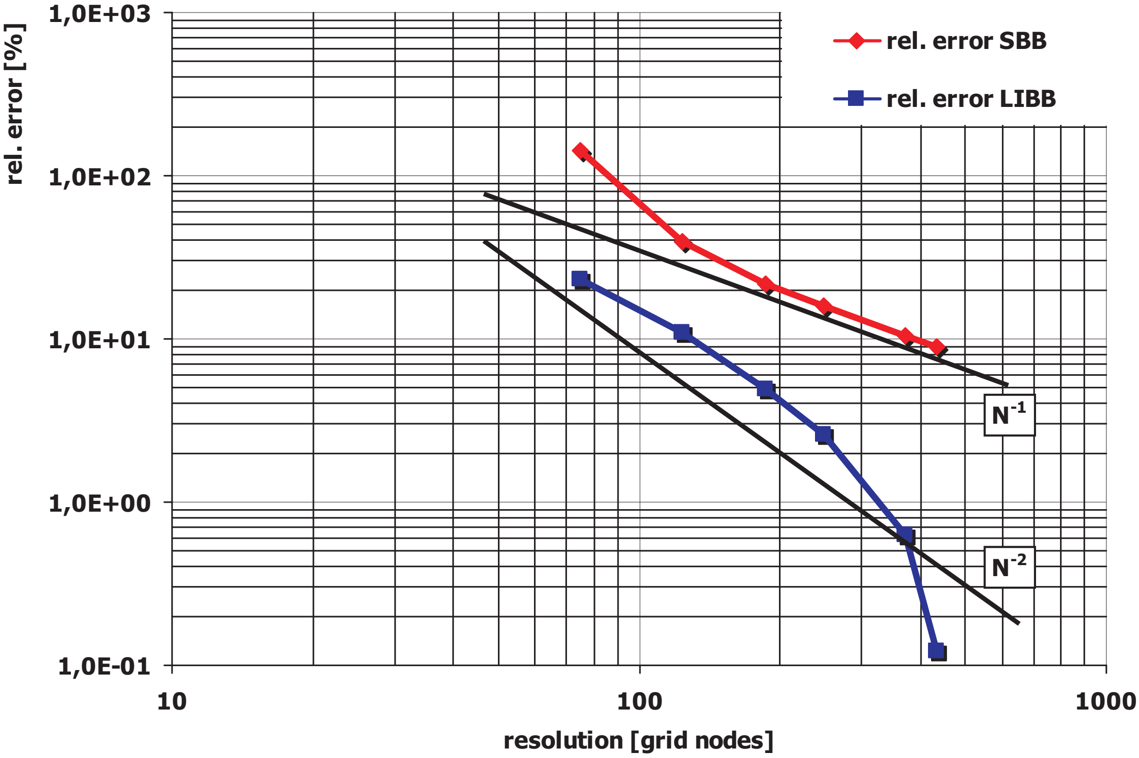


Figure 6.4.1 – Plot comparing absolute permeability depending on the resolution.

complex multiphase simulations the balance between parameters that yield reasonable results and numerical efficiency has to be carefully explored.
Besides the forces mentioned above the history of flow plays an important role. Thus e.g. drainage and imbibition show a hysteretic behavior. This effect is described in the following section.

6.6 Hysteresis

Hysteresis at the pore scale is driven by up to four effects: the "Ink Bottle“-effect, contact angle hysteresis, accessibility, or network effects as well as aging. The next paragraphs will describe the basic concepts of the effects causing a hysteretic behavior in porous media.

Ink Bottle effect

The basic idea of the Ink Bottle effect is that if pores have a large chord length or pore body which then connects to neighboring pores through constricted pore throats, there is some hysteresis already at the pore scale. It is assumed that a pore is initially filled with wetting phase w and the interface to the non-wetting phase nw is positioned at the left opening as shown in Figure 6.6.1. During the drainage process the pore is evacuated until the interface of the non wetting phase reaches the pore throat at the opposite site of the entry point. Now significant less capillary pressure is needed to displace the pore throat by non wetting phase (see Figure 6.6.1).

However, if the process is reversed during imbibition, the capillary pressure needed to re-wet the pore volume has to be higher than for the drainage (compare the last stage on the right in Figure 6.6.1). Such discontinuous changes of fluid content observed in capillaries are referred to as Haines jumps [68].

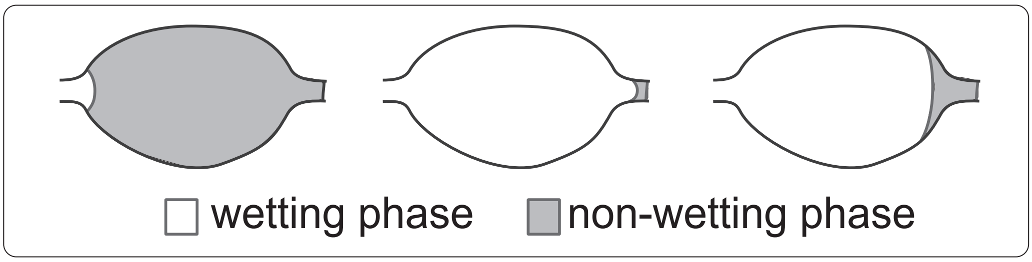


Figure 6.6.1 – Ink Bottle effect contributing to Hysteresis.

Contact angle hysteresis

Contact angle hysteresis is a well-recognized phenomenon, but it is only significant at relatively high flow rates. The slower the process, the less contact angle hysteresis. During the drainage process, if the non-wetting phase displaces the wetting phase, the advancing contact angle is smaller than the receding one during the reversed process - the imbibition. The hysteresis is mainly caused by surface roughness, chemical contamination or heterogeneity of the solid substrate, and deposition of solutes from the liquid onto the solid surface.

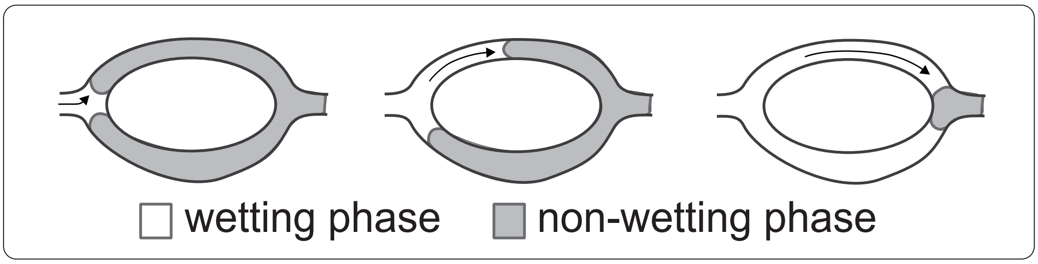


Figure 6.6.2 – Slow by pass effect contributing to Hysteresis.

Phase trapping

There are two effects leading to phase trapping: the snap-off effect and the by-passing effect [29]. While the by-passing effect can occur both during drainage and imbibition, the snap-off effect mainly happens during the imbibition. If one phase displaces another in a material dominated by capillary connections of different diameter classes the displacing phase advanced faster in

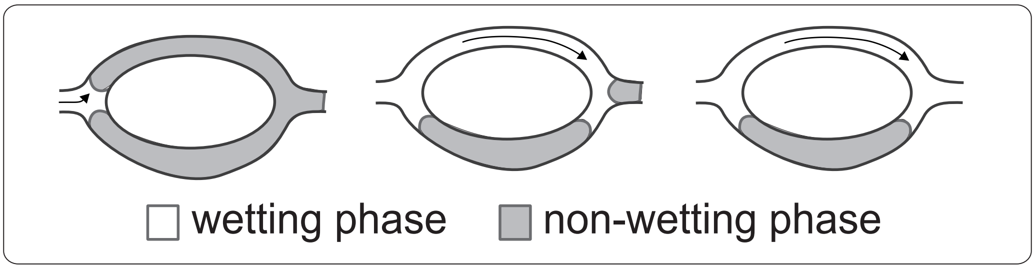


Figure 6.6.3 – Fast by pass effect contributing to Hysteresis.

capillaries with a larger diameter then in capillaries with a smaller one. Therefore the much faster advancing phase in the wide capillary surpasses the slowly progressing receding phase in the narrow capillary and triggers a disconnect of the receding phase which then leads to phase entrapment.
During the snap-off blobs of the displaced non-wetting phase can be held back, depending on the contact angle and on the relationship between diameters of pore cavities and pore throats [158]. This effect is illustrated in Figure 6.6.4. The combination of these two effects are sometimes also called the accessibility criterion.
A typical drainage and imbibition curve is illustrated in Figure 6.6.6 where the capillary pressure-saturation relationship is affected by hysteresis and therefore the characteristic loop is developed.

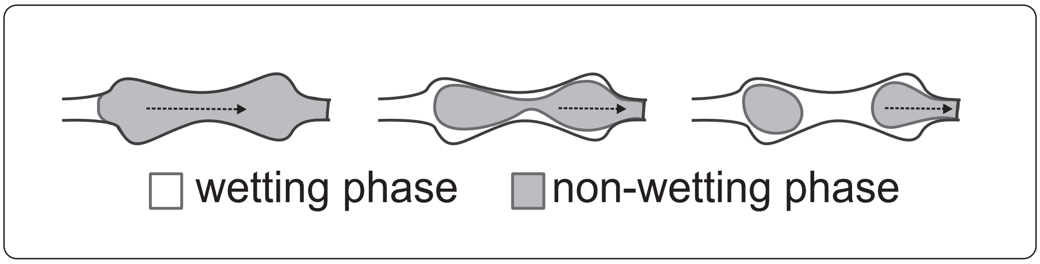


Figure 6.6.4 – Snap-off effect in a capillary with varying radii.

Aging effects

Aging effects are strictly spoken not contributing to hysteresis, however it may be listed due to completeness and as an explanation why 100% saturated soil can indeed exist in nature. During imbibition, air can become trapped inside the larger soil pores as already explained above. Given sufficient time, the entrapped air bubbles will slow dissolve in the water and diffuse to the edge of the sample, where they eventually return to the atmosphere. This process takes a long time compared to the other effects causing hysteresis in porous media. Aging effects are

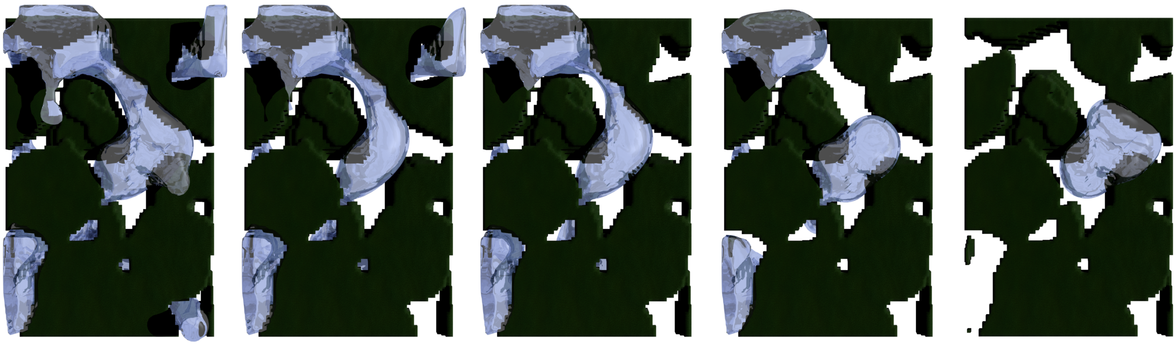


Figure 6.6.5 – Snap-off effect in a sub section of a porous material.

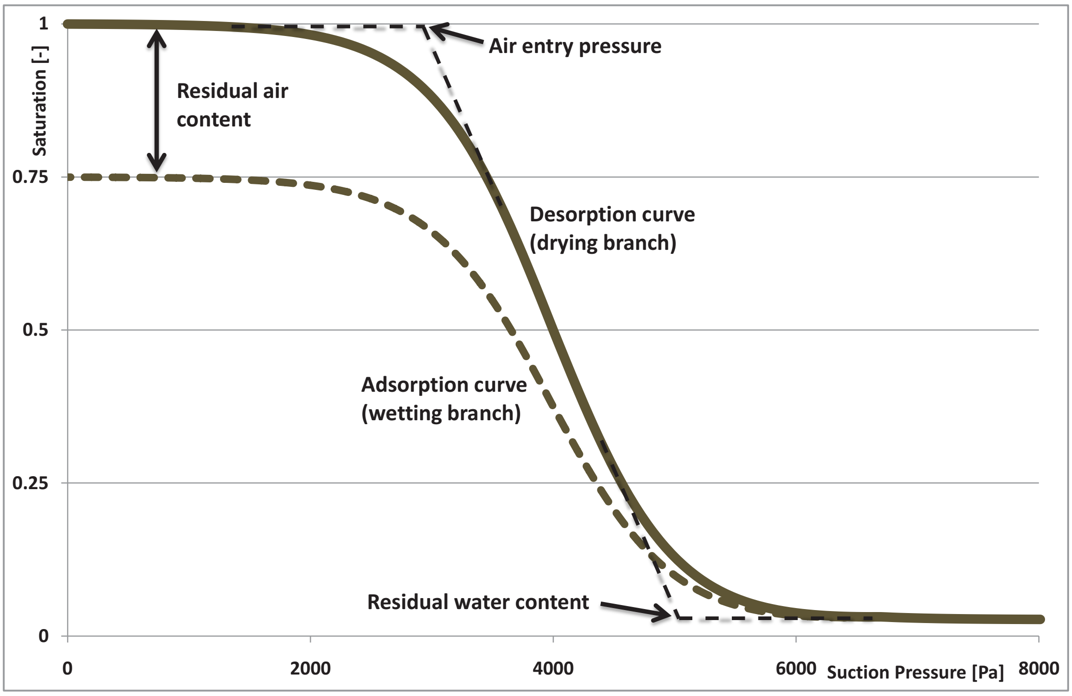


Figure 6.6.6 – Typical drainage and imbibition curve caused by hysteretic effects.

not considered in the numerical experiments, but they may be crucial for the initial saturation process while performing physical measurements in porous media.

6.7 $P_c - S_w$ relationship

Only a few studies [1, 2, 11–13, 101, 150, 164] have reported microscopic simulations of multiphase flow in three-dimensional porous medium systems, in part because of computational limitations. Computations of the $P_c - S_w$ relationship based on lattice Boltzmann simulations can be found in [95, 113, 150, 157]. The computations carried out for $P_c - S_w$ relationships in [95, 150] have been performed with the standard Gunstensen model. They are of qualitative nature and the results were not grid independent. In [157] the effective grid resolution was increased, but only the primary drainage curve has been computed and compared to results from other simulation methods[5] , but not to experimental data. [113] calculated the hysteresis of capillary pressure-saturation relationships using the Shan-Chen model and compared them to experimental data obtaining good results. One drawback of the method used by [113] is that the primary physical parameters as the fluid-fluid and fluid-solid interaction coefficients must be determined by a model calibration using numerical experiments. Also, due to a limited range of stability, the time step is very restrictive. During the numerical experiments the three sub-samples A_{Min}, A_{Max} and A_{Med} for the computation of the hysteresis have been used. Hereby a model calibration was not necessary and the capillary pressure-saturation relationship has been computed by using a system without gravity. Initially the entire pore space is filled with wetting phase (i.e. water). To compute the $P_c - S_w$ relationship including hysteresis a time dependent pressure difference is applied. At the top of the sample a non-wetting phase reservoir is given with a constant reference pressure $P_0 = 0$. At the bottom of the sample a time dependent decreasing pressure $P(t)$ is imposed. The static $P_c - S_w$ relationship is a curve defined for an infinite number of pressure steps and $Ca \rightarrow 0$. In principle this could be performed by discrete pressure jumps and waiting for a steady state after each jump. But this becomes very tedious for a large number of pressure jumps, so a linearly increasing and decreasing pressure boundary condition is used to compute the primary drainage, the first imbibition and the secondary drainage:

$$P(t) = \begin{cases} -P_{max}\frac{t}{T}, & \text{if } t \in [0, T[\\ -P_{max}(1 - \frac{t-T}{T}) & \text{if } t \in [T, 2T[\\ -P_{max}\frac{t-2T}{T}, & \text{if } t \in [2T, 3T] \end{cases} \tag{6.7.1}$$

The maximum pressure load P_{max} should be chosen in a way that the smallest pores can be evacuated and therefore estimated by equation (4.10.1). To obtain a sufficiently slow process the time T has to be large. The time scale order has been estimated by using the equation derived by Washburn [159], here modified for a linearly increasing pressure from 0 to $P = \frac{2\sigma}{r}$

$$\Delta T_{ref} = \frac{4L^2(\mu_w + \mu_n)}{\sigma \, r_{ref}}. \tag{6.7.2}$$

ΔT_{ref} gives an approximation for the time a two phase system needs to penetrate a distance L into a fully wettable, porous material whose average pore radius is r_{ref}. Now one can integrate over the pore radius from r_{small} to r_{big}, where r_{small} is related to P_{max} by $P_{max} = 2\sigma/r_{small}$ and r_{big} is an approximation of the largest pore size. Estimations about T_{int} can be obtained for the time T in equation (6.7.1):

$$T_{int} = \frac{1}{\sigma} 4L^2(\mu_w + \mu_n)(\ln r_{big} - \ln r_{small}) \tag{6.7.3}$$

If one assumes that the smallest resolvable radius is defined by 2 grid spacings and the largest by approximately 20 grid spacings, $T_{int} \approx 2600L\,\mu_w/\sigma$ is obtained. Regarding numerical efficiency, one can assume from Equation 6.7.3, that for a small T_{int} a low dynamic viscosity and a high surface tension is favorable. The dynamic dependence of the hysteresis on the time T for the sample A_{Med} is explored by choosing different times T. The parameters for the setup have been:

---

[5] In [157] comparisons have been performed against a morphological pore network model which is similar to the model basically described in section 6.1.

$\mu_n = 1/200$ [Nsm^{-2}], $\mu_w = 1/500$ [Nsm^{-2}], $\sigma = 1/100$ [N/m] and the maximum pressure was $p_{max} = \sigma\,/\,r_{ref}$ [Pa], where $r_{ref} = L/200$ [m] with L as the size of the domain. The ratio M of the dynamic viscosities is $M = 2/5$, which is far from the real world value (air/water) 1/50. This was mainly chosen to improve numerical efficiency, since for a smaller ratio the computational times have become very large.
For smaller grid sizes the effect of M has been investigated and found that no important effect for this type of media occurred, even when using larger values of M. The numerical grid size was chosen as 200^3. In Figure 6.7.1 the results for different fast processes are shown. For the simulations A-D T was set to (1250, 2500, 5000, 10000) $\times L\,\mu_w/\sigma$ respectively, resulting in ($0.125{\times}10^6$, $0.25{\times}10^6$, $0.5{\times}10^6$, $1.0{\times}10^6$) LB iterations per branch. The Figures show a strong dependence of the hysteresis on T, but one can also see a convergent behavior for the more saturated range of the retention curves and for the residual air saturation. In the dry range and for the residual saturation one can see a strong dependence on T. For simulation D an almost residual saturation of 0 (film flow) is obtained, indicating a resulting static process. The intersection of the curve in the dry range is a numerical artifact and is related to the insufficient resolution of thin films in the LB method. For subsequent simulations $T = T_D$ (as in Equation 6.7.1) is used. In Figure 6.7.2 the hysteresis for the minimum and maximum geometry

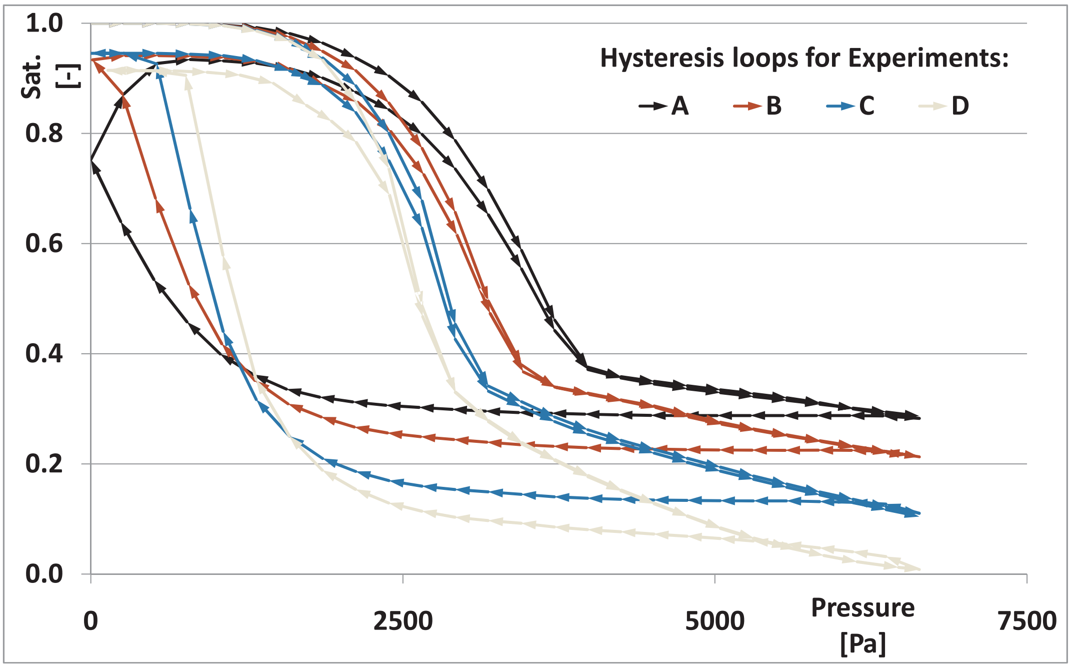


Figure 6.7.1 – LBM simulation (A_{Med}) of hysteresis by different fast processes.

A_{Min} and A_{Max} are shown. The air-entry value is larger for A_{Min} and the hysteresis is much more pronounced. The maximum liquid saturation is quite similar for both geometries and is about 0.9. In Figure 6.7.3 the residual non-wetting phase after the imbibition for A_{Min} and A_{Max} is shown.

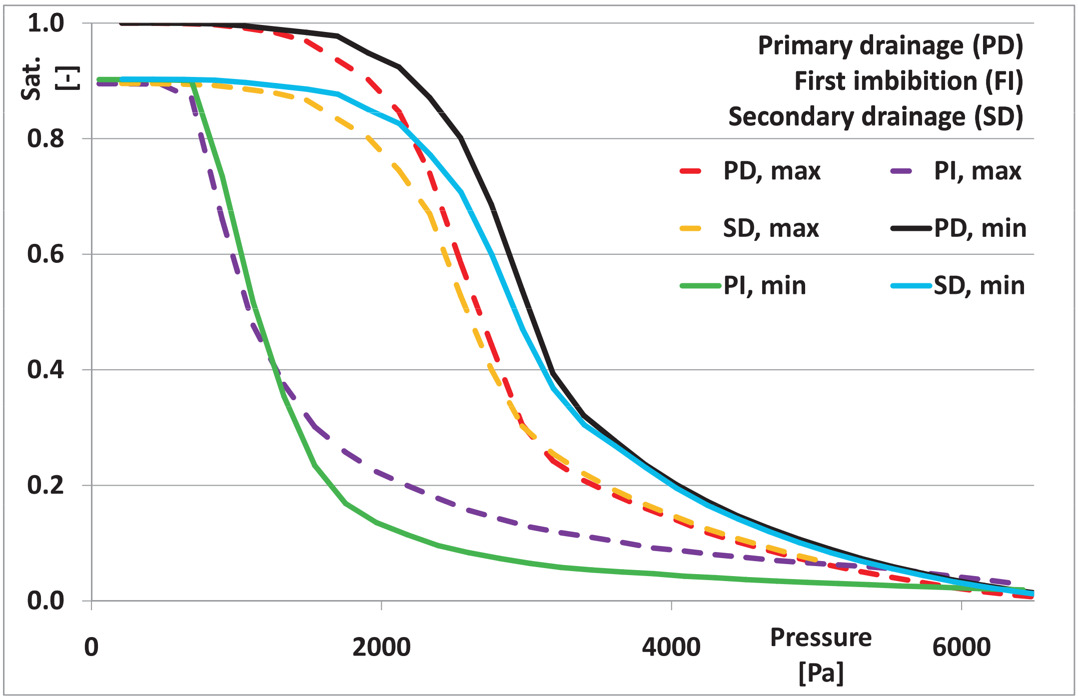


Figure 6.7.2 – LBM simulation of hysteresis for different geometries (A_{Min} and A_{Max}).

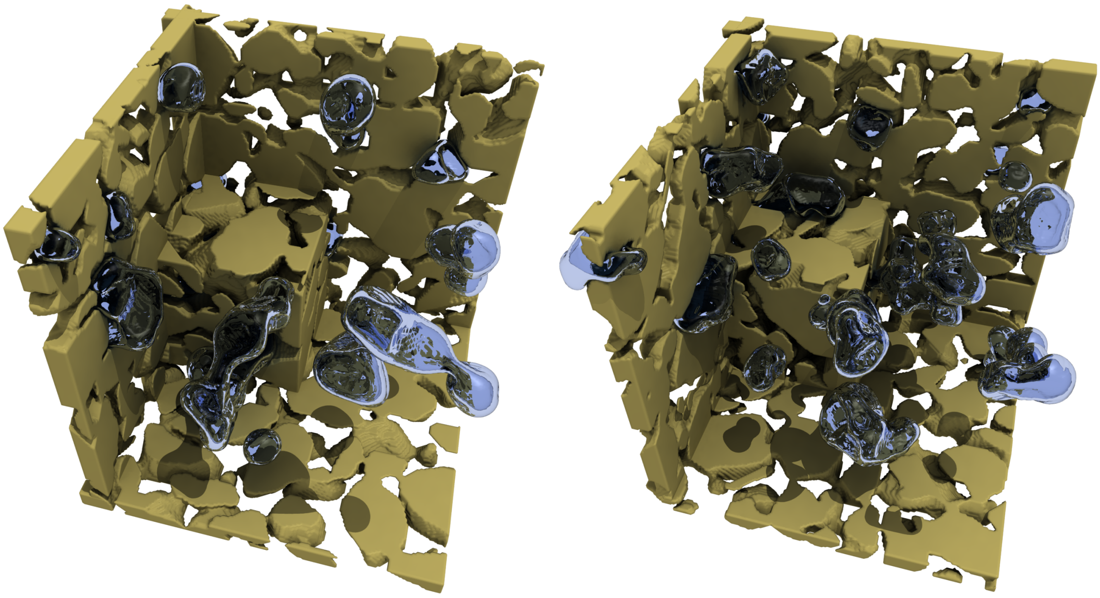


Figure 6.7.3 – Residual air saturation after the imbibition, left: maximum geometry A_{Max}, right: minimum geometry A_{Min}.

6.8 Comparison of results

A brief comparison of experiment, LB simulation, and MPNM approach is showed in this section. For a more detailed description as well as for secondary literature see [7] and the references therein.

In Figure 6.8.1-6.8.3 the $P_c - S_w$-relationship for the primary drainage, the first imbibition and the secondary drainage are shown. The results are compared between the experimental data, by the morphological pore network model (see section 6.1) and the lattice Boltzmann model (see section 6.7). The experimental results obtained from column 1 and 2 are fitted using either a constrained van Genuchten model [154] (vG) or free form Hermite spline functions (FF) as in [18]. Details to this procedure as well as further fitting methods can be found in [7]. While the experiments were performed on 7.3 cm high columns of packed sand with a diameter of 9.4 cm, the numerical experiments have been executed using a 200^3 voxel sub sample (A_{Med} see section 6.3) of a CT scan of the very same sand. For the primary drainage the results of the LB simulation and the morphological pore network model show a very good agreement. Only in the dry range, where the LB model suffers from the problem of the film flow (discussed in section 4.10) and resolution problems of the wetting phase (discussed in section 4.6), deviations occurred. The results of the inverse fits from column 1 and 2 are very similar. They deviate from the results of the LB- and MPN-methods in amplitude by approximately 10%. The value of the slope agrees very well. The residual saturation predicted by the MPN model is 12% and the inverse fit predicts values from 15 to 18%. For the first imbibition the results of the inverse fit from column 1 and 2 are different. Column 1 showed a very late response leading to a step function for the free form fit, but the van Genuchten type is balancing this behavior, see Figure 6.8.2. The results for column 2 are distinctly smoother. Nevertheless the results obtained by either the LBM or the MPNM show more resemblance to the results from column 1. In the saturation range between 0.2-0.7 the LB and morphological results have a similar slope but the amplitude of the latter is 25% higher. The maximum water content for the wetting process is 84-86% for the experimental, 76% for the morphological and 90% for the LB results. For the

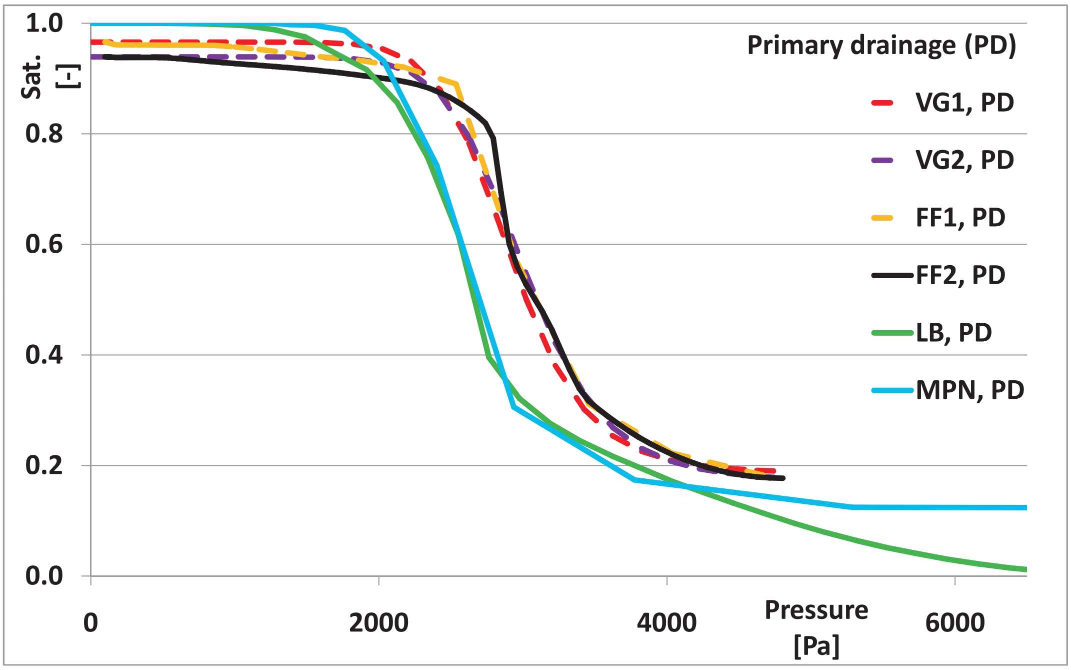


Figure 6.8.1 – Primary Drainage (PD): Morphological pore network model, LBM simulation and parameter fits vG and FF.

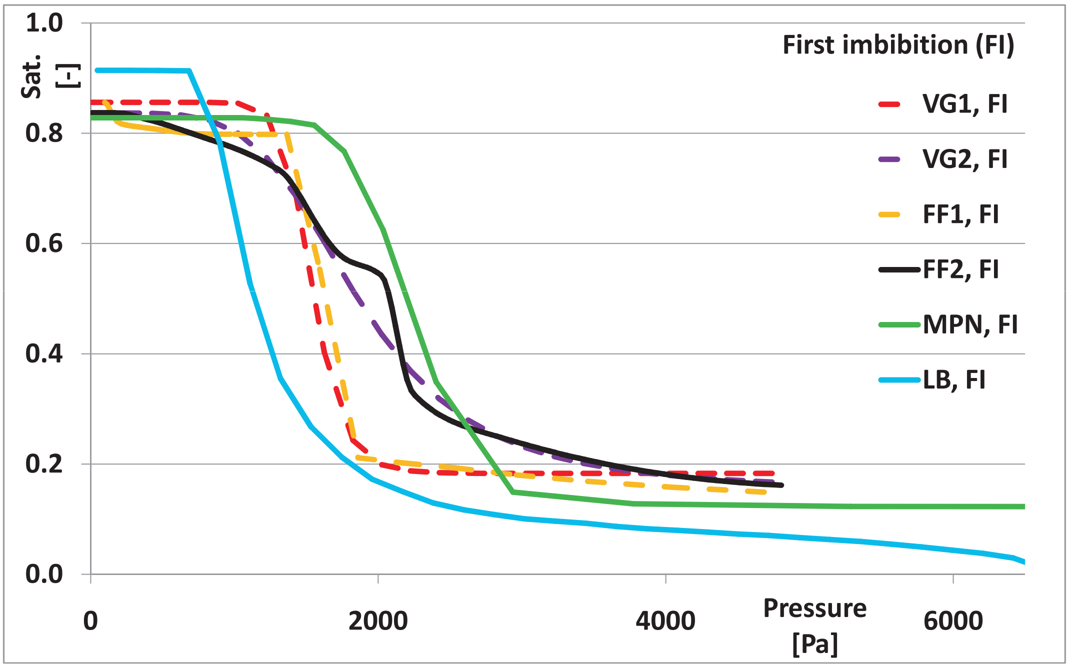


Figure 6.8.2 – First Imbibition (FI): Morphological pore network model, LBM simulation and parameter fits vG and FF.

secondary drainage the results from the LB simulation and the MPN model show a very good agreement in the range of saturations between 0.6 and 0.15. In the wetter range the difference is affected by different residual air saturations from the first imbibition. In the dry range the LB method drains almost completely due to film flow. The results for the inverse fits from column 1 and 2 are very similar. They deviate from the results of the LB- and MPN-model in amplitude by approximately 10%. Again the value of the slope agrees very well. In Figure 6.8.4 the hysteresis for the minimum geometry, A_{Min}, in Figure 6.8.5 the hysteresis for the maximum geometry, A_{Max}, obtained from the LB simulation and MPN method are shown. They show good agreement for the drainage curves, yet for the imbibition the deviations are larger. The air entry pressure for A_{Min} is higher than for A_{Max} and this behavior is reproduced by both models. This can be explained by the smaller throats of the pore space in the minimum geometry (small pore width and porosity) that need a higher pressure to be drained. The maximum water content for the wetting process is approximately 0.9 in both cases for LB and 0.75 (A_{Min}) and 0.78 (A_{Max}) for the morphological pore network model. Referring to the LB results, the hysteretic effect is much more pronounced for the minimum geometry than for the maximum geometry. We observe the same behavior in weaker form for the MPNM as well.

In this section the capability of two different multiphase models has been evaluated in order to predict the hysteretic behavior of the capillary pressure saturation relationship on the basis of detailed representations of the complex porous structure as measured by X-ray tomography. The models differ considerably in complexity and computational costs and can have different potential applications. However, the results obtained for the primary and secondary drainage curve are consistent, while differences among the models for the first imbibition are observed.

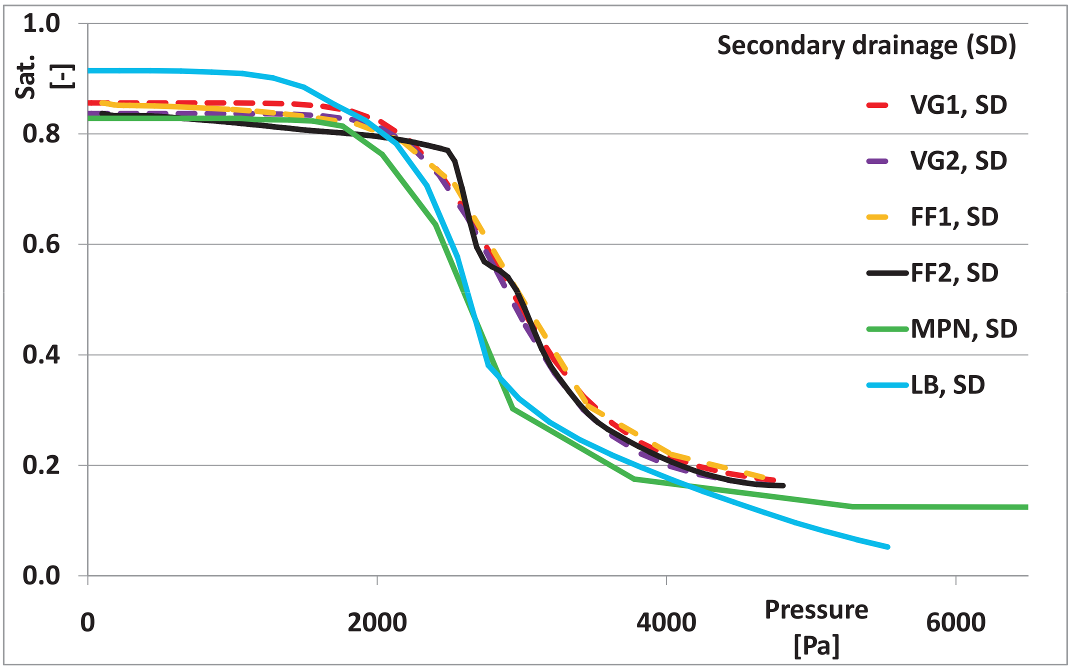


Figure 6.8.3 – Secondary Drainage (SD): Morphological pore network model, LBM simulation and parameter fits vG and FF

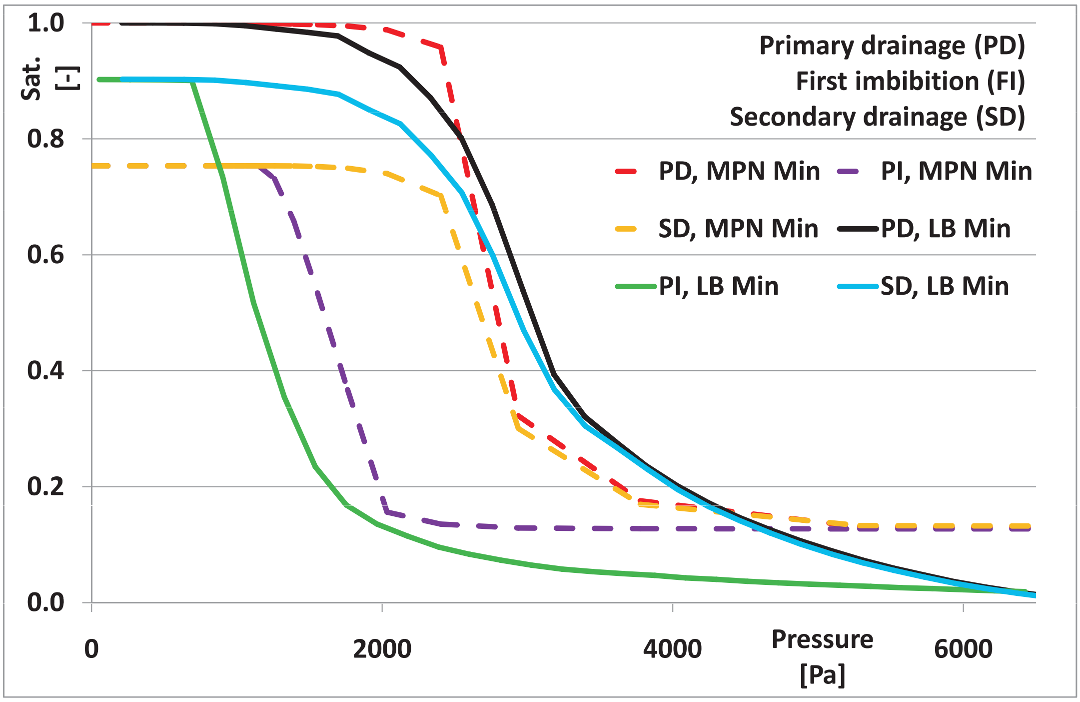


Figure 6.8.4 – A_{Min}, LBM and MPNM results.

In general MPN models are sufficient to compute the primary drainage, but the choice of the structural element for imbibition is not straightforward and changes the results considerably. In

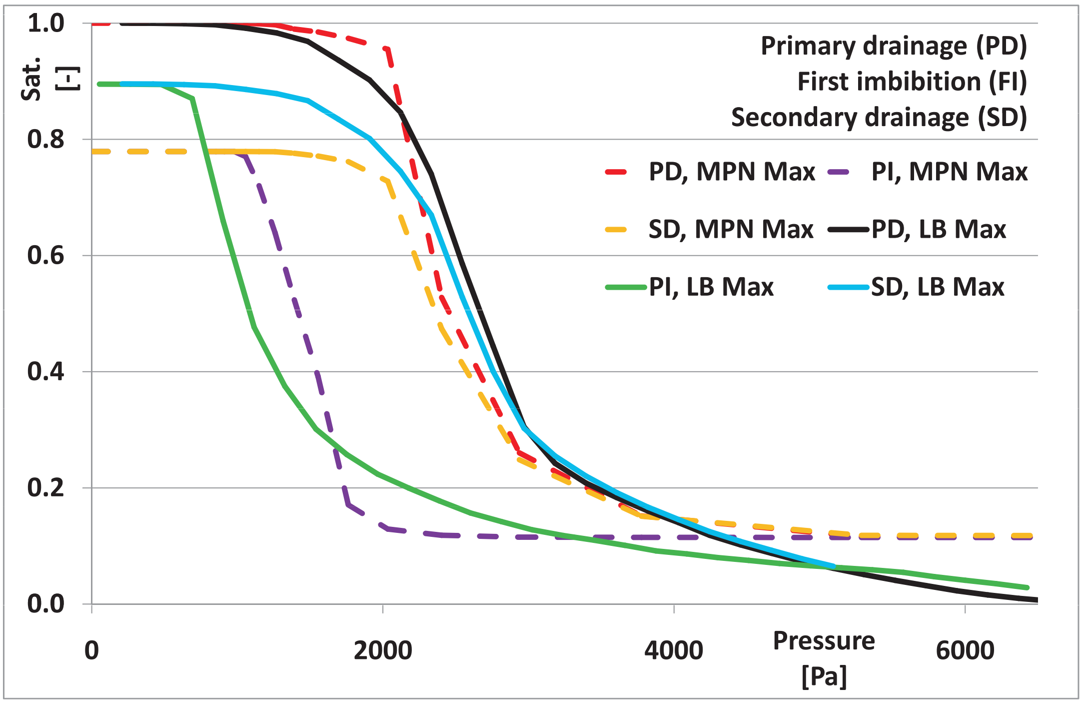


Figure 6.8.5 – A_{Max}, LBM and MPNM results.

contrast the LB approach needs no modeling for the imbibition, but due to a lack of resolution in the dry range, numerical errors become large and reliable results are difficult to obtain.

7 Thermal and acoustic applications

The advection/diffusion extension for the lattice Boltzmann method introduced section 2.3 has been used to simulate various problems, which were related to different fields of interest. Examples include the transport of contaminants in groundwater, sediment transport processes and miscible fluid flow problems in various contexts. The simulation kernel introduced has the capability to simulate advection/diffusion problems of variuos kinds (see section 2.3), however, it is beyond the scope of this thesis to pursue e.g. the simulation of moisture evaporation from porous media as illustrated in Figure 7.0.1.

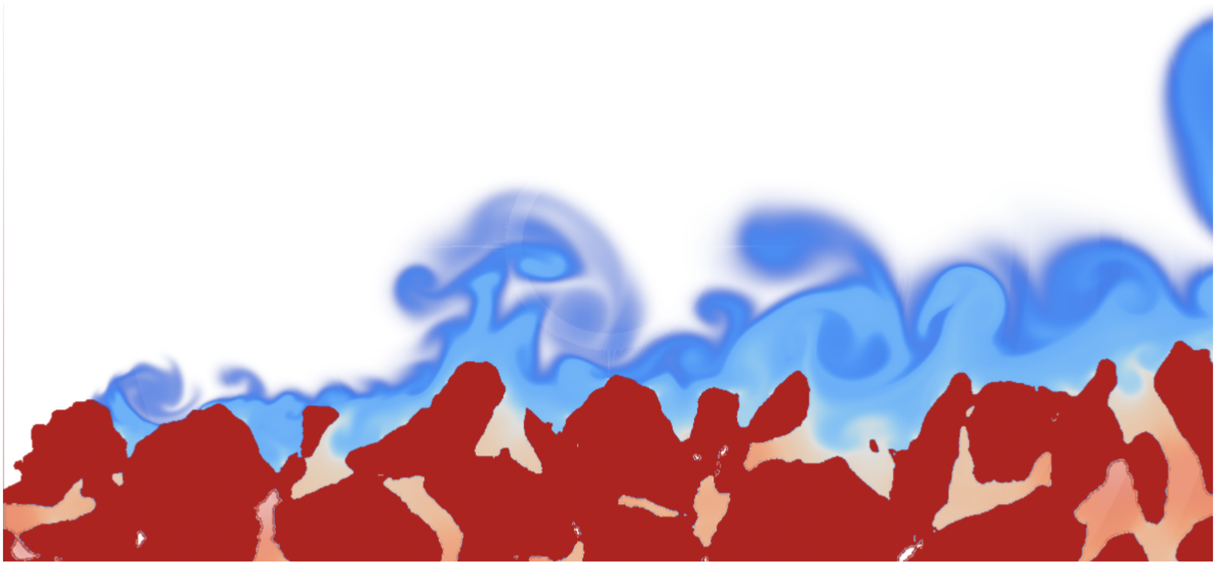


Figure 7.0.1 – Simulation of moisture evaporation of a porous surface dominated by advection

Applications using smoke or fume, or thermal properties are also considered advection/diffusion problems. The thermal properties of the extension to the lattice Boltzmann model introduced in section 2.4 are briefly investigated in this chapter, but a more detailed description of the same model extension, where the temperature (energy) equation is solved using an additional finite difference scheme, can be found in [145] and [155].

The attenuation of sound waves in the context of an LB simulation has been discussed in section 4.11. As an application example, the absorption of sound waves in a passive porous absorber has been investigated using both uniform and non-uniform grids [50].

Therefore, this chapter is only a brief and quantitative overview of engineering applications computed with the VFC and the VF framework.

The applications presented in the following sections demonstrate the versatility of the lattice Boltzmann method and its extensions. However, it is beyond the scope of this thesis to specify the numerical approaches in more detail.

7.1 Thermal flows

Thermal flows, which are closely related to advection/diffusion problems, arise in many engineering applications (e.g. the simulation of heat exchangers, turbomachines, 'Heating, Ventilating, and Air Conditioning' (HVAC), fire, forced and natural convection problems, etc.). Using the VFC framework and the integrated Hybrid Thermal LBE (HTLBE) approach introduced in section 2.4, this section exemplarily demonstrates the benefits to apply the simulation package in computing a forced convection inside a buildings double fassade. The aim of this simulation was to estimate the efficiency of three different versions of fassade constructions (see Figure 7.1.1).

The simulation of the three different variants have been performed assuming worst case scenarios of the weather conditions (35° Celsius outside temperature, no wind). In order to get reasonable input data relative to the initial temperature distribution and the insertion of thermal energy into the system, a radiative heat transfer simulation [17] has been performed. Here, the surface temperature of all materials exposed to direct sunlight is determined and used as

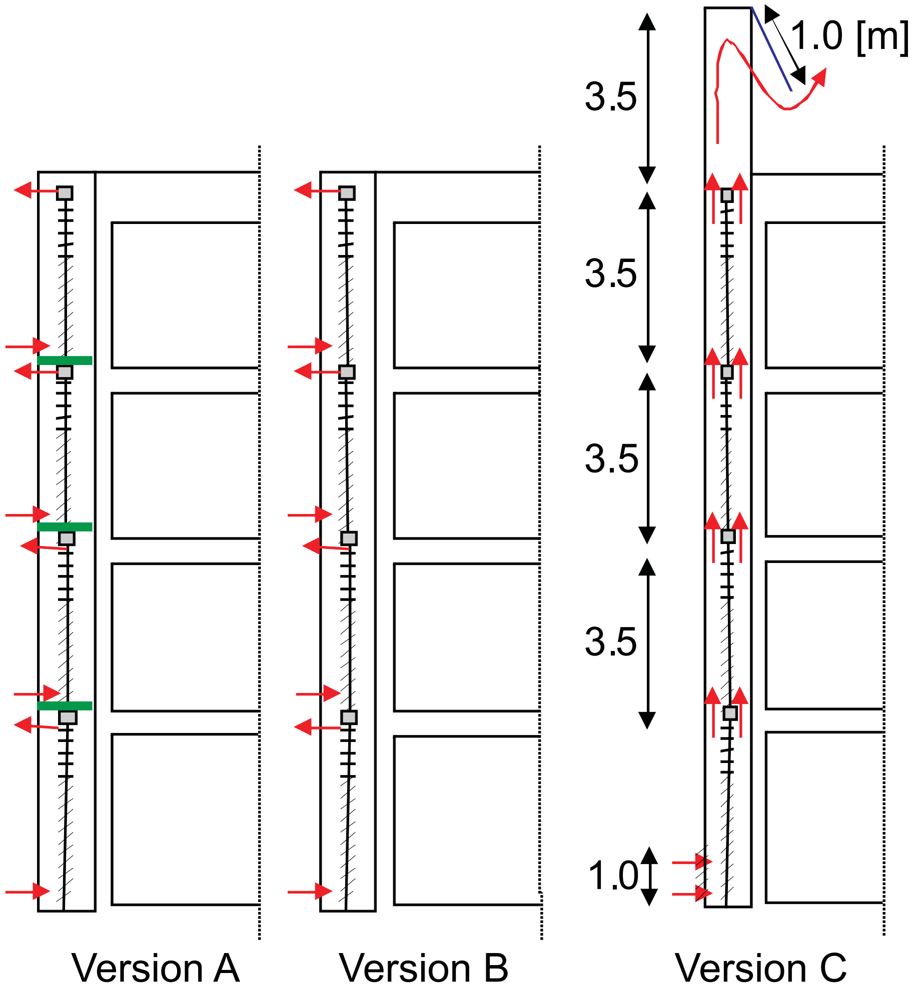


Figure 7.1.1 – Design of a buildings double fassade in three different versions: partitioned per structural level, continuous with in- and outflow per level, and continuous with exhaust air flap above top level.

input data for LB simulations.

To get a better understanding of the convective heat-transfer inside the simulated system and

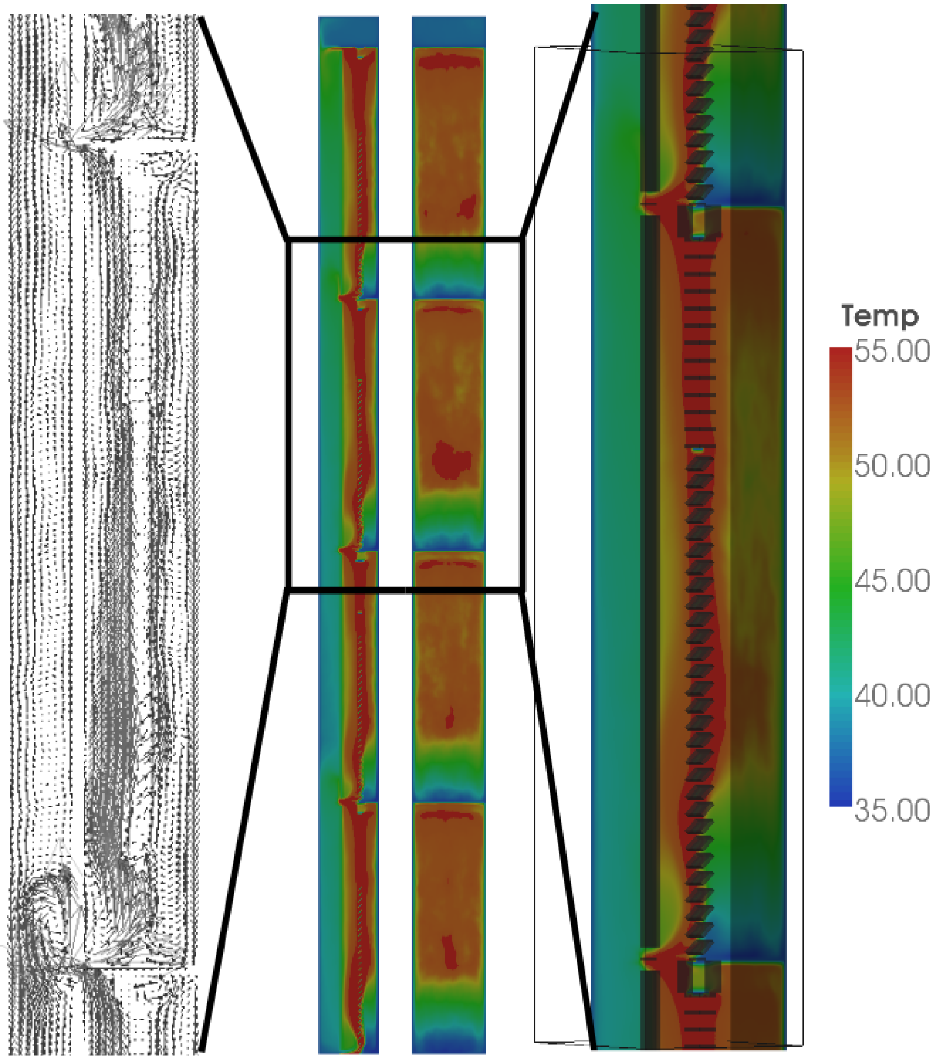


Figure 7.1.2 – Orthogonal cut planes of temperature distribution and velocity in fassade type A

to compare the numerical results to the real world, some basic assumptions have been made. To estimate the validity of the approach, a basic comparative analysis has been performed, using heat transfer correlation between a turbulent fluid and a plate. The heat exchanged (heat flow) between fluid and plate is expressed as [114]:

$$\dot{Q} = |\alpha A \Delta T| \tag{7.1.1}$$

where A is the surface area, ΔT the temperature difference between environmental and surface temperature and α the heat transfer coefficient. The Nusselt number Nu, which is the ratio between convective and conductive heat transfer across the boundary, is related to α as:

$$\alpha = \frac{Nu\lambda}{L} \tag{7.1.2}$$

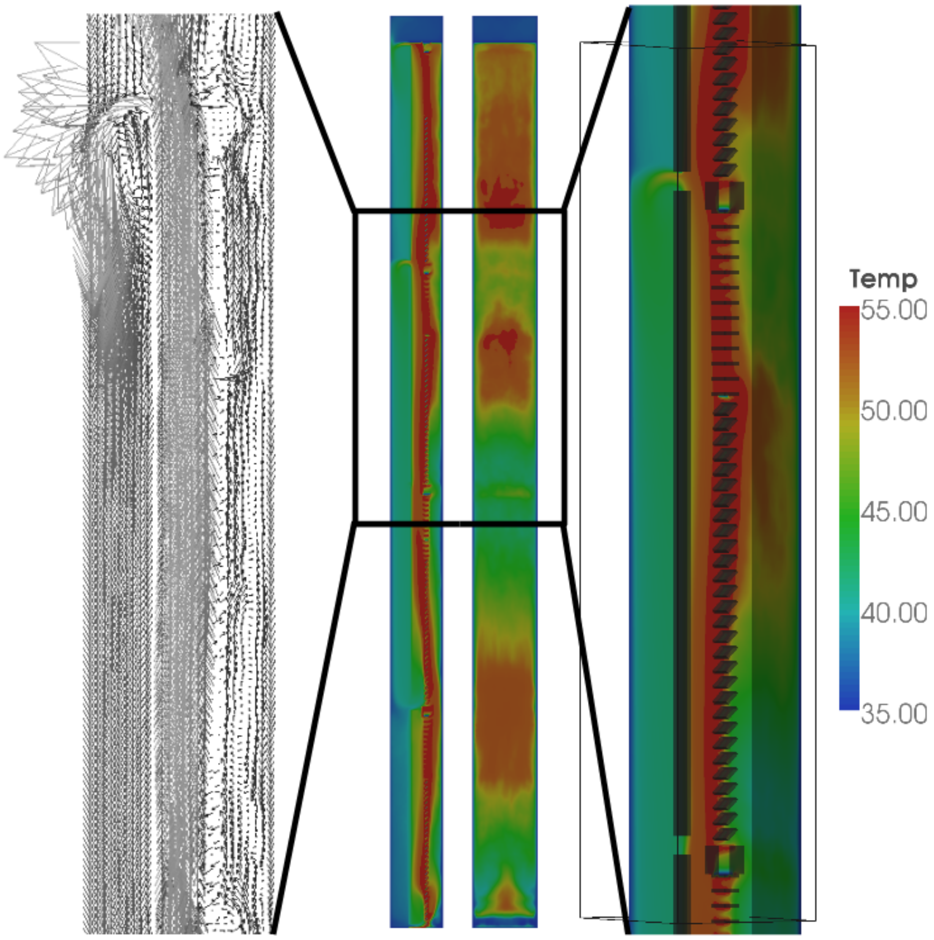


Figure 7.1.3 – Orthogonal cut planes of temperature distribution and velocity in fassade type B

where L is the characteristic length of the plate and λ the thermal conductivity of the fluid. The Nusselt number is, with a good approximation, expressed as:

$$Nu = 0.0296\mathrm{Re}^{0.8} Pr^{1/3} \tag{7.1.3}$$

in an interval of $\mathrm{Re}_{crit} < \mathrm{Re} < 10^7$ and $0.6 < Pr < 15$, where Re is the Reynolds number and Pr the Prandtl number, which is 0.71 for air. Assuming a temperature difference of $\Delta T = 25°$ Kelvin, a Reynolds number of $\mathrm{Re} \approx 55000$ (with the depth of the fassade as characteristic length) and a surface area of the plate of 3.5 m^2 the resulting heat flux is ≈ 540 W. This value corresponds very good to the approximately estimated heat flux obtained from the simulation results in every structural level. Therefore, model type and simulation setup permits a qualitative conclusion which structural design is favorable in terms of cooling efficiency.

7.2 Acoustics

Vibrating bodies, regions of turbulent flow, the mixing of flows of different temperatures, and other unsteady phenomena produce fluctuations in pressure that propagate through a real fluid as sound. Sound waves are nondispersive in a homogeneous fluid over a broad range of frequencies, for which viscous and thermal diffusion are small (this is particularly true at audible

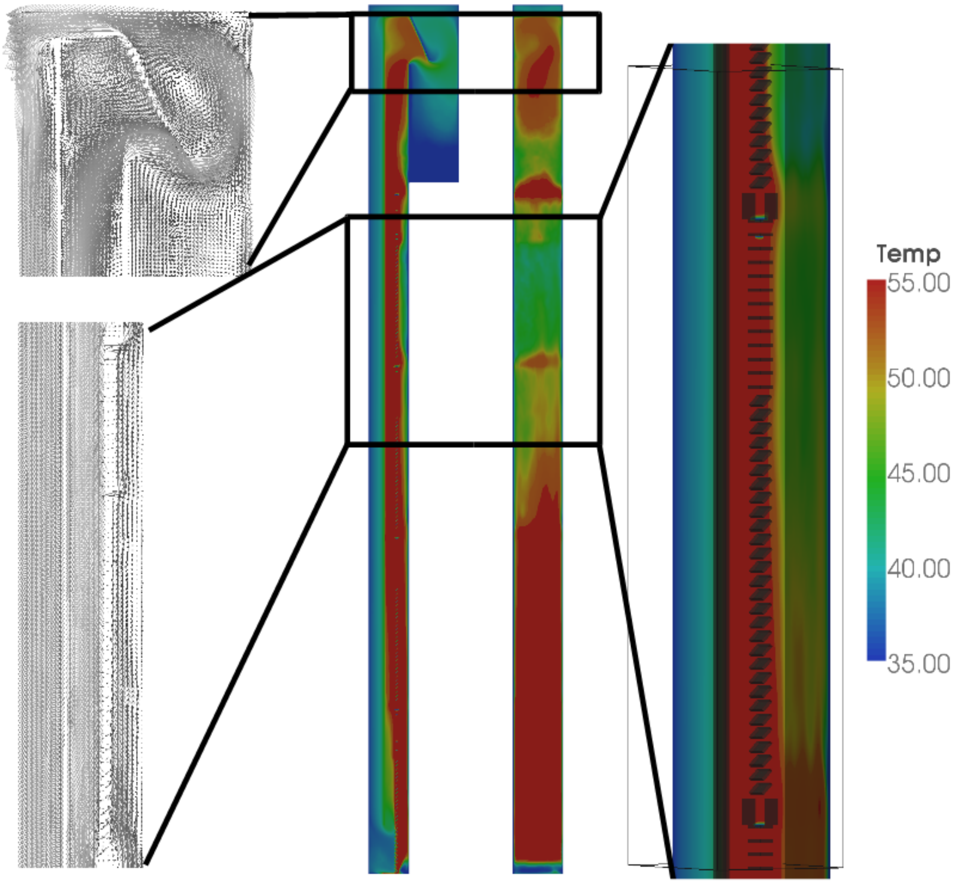


Figure 7.1.4 – Orthogonal cut planes of temperature distribution and velocity in fassade type C

frequencies), and it ensures that pressure fluctuations representing, e.g. speech, remain coherent over large distances.

While focusing on sound attenuation in porous media, two types of validation experiments have been conducted (see section 4.11). First and most basic the attenuation of a sound wave in an unbounded fluid has been observed, finding very good agreement with analytical predictions, and secondly the damping of an acoustic wave confined in a tube have been considered.

Sound absorption in porous media, however, depends on various physical mechanisms. Besides the viscous losses at the inner surface wall inside a porous medium, there are other effects, which have to be considered [104, 110], such as: resonance, coupling of structure and fluid, and thermal losses. Theories about attenuation of sound in complex structures are partly unable to describe the influence of the mentioned effects. The simulations performed using the lattice Boltzmann flow solver includes the viscous losses, as well as the resonance effect. However, the transition of acoustic energy into the porous material is not modeled yet (fluid/structure interaction) and indeed a challenge. Furthermore, the thermal losses, which are considered to be very small, are not included in the present LB Model.

The setup which has been used to compute the attenuation of sound in porous media is as follows: a subsection of a sample of porous asphalt has been uniformly discretized using the grid generation process described in chapter 5. The sample has been obtained by tomography methods and consists of grey values, keeping the geometry second order accurate in space. Furthermore, due to the high quality of the scan it is possible to separate the different materials, like mineral, bitumen and pore space. The iso-level threshold has been set between pore space and bitumen. The subsection used for the numerical grid consists of at least five pores and grains in every spatial direction to ensure an acoustic REV. Above the pavement surface a bulk fluid, with a height of a corresponding frequency range from 100 Hz up to 2500 Hz, has been initialized. Using uniform grids, this setup is very time consuming, because the porous media itself is resolved by approximately 250^3. The voxel size has been 0.0001 m per node, therefore the corresponding LB wavelength of 1000 Hz is 3450 lattice spacings, and for a frequency of 100 Hz, even worse, one order of magnitude higher: 34500 lattice spacings. This is an ideal case for applying a grid coarsening, respectively a grid refinement approach, such as provided in VIRTUALFLUIDS [49]. Here, the asphalt is resolved at the highest level, but still at 250^3 grid nodes, the bulk fluid is initialized at the coarsest level (see Figure 7.2.1).

Periodic boundary conditions are used in the bulk fluid and a sinusoidal pulsating pressure boundary condition, inserting a sound wave into the fluid, is applied above the porous pavement sample. The sound wave is reflected by a wall located underneath the porous material.

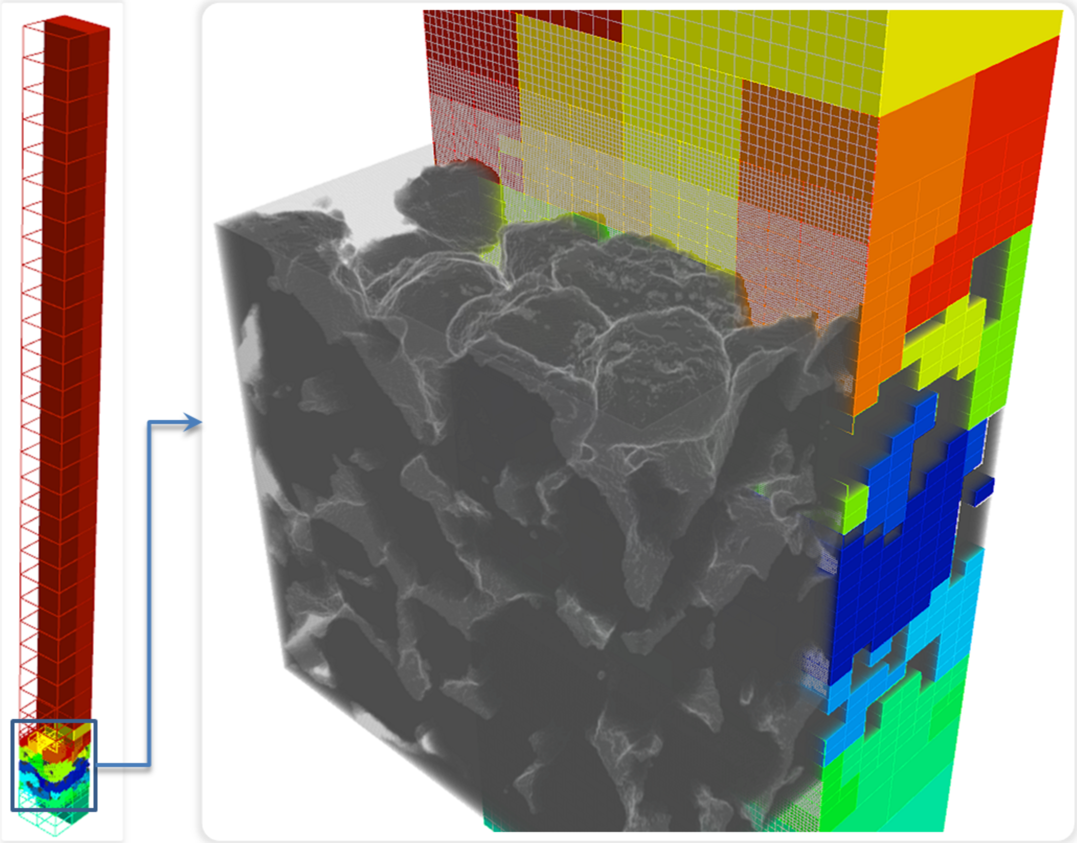
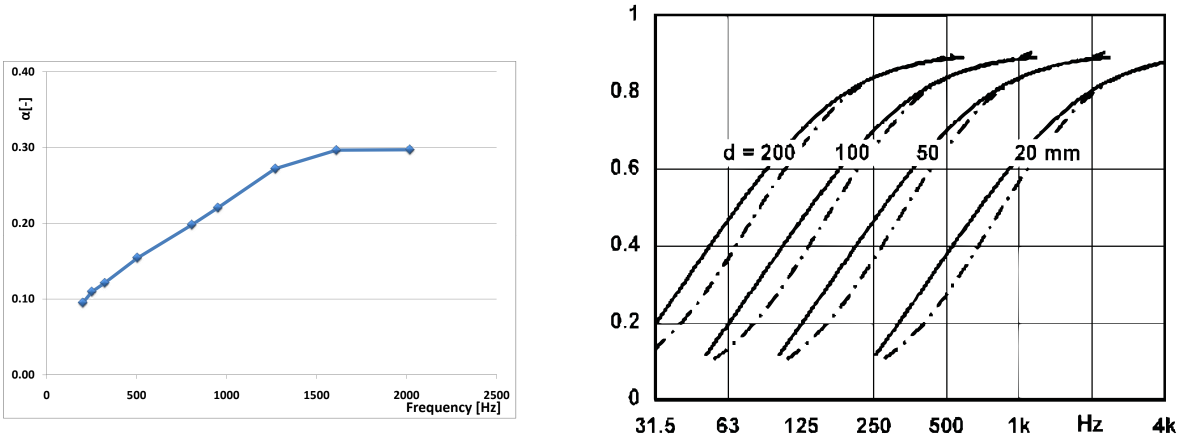


Figure 7.2.1 – Refined grid around a porous asphalt to resolve the acoustic boundary layer (picture courtesy of [49])

The numerical experiment has been conducted by transmitting one period of a certain frequency inside a range of 100 to 2500 Hz, measuring the incoming and the reflecting wave. The results illustrated in Figure 7.2.2a and in Figure 7.2.2b shows a typical absorption diagram for a porous material for comparison. The resulting absorption curve of the LB experiment is in shape similar to the reference curve, however the amplitude deviates strongly. This is due to the missing resonance effect inside the pores[1] and the absence of a fluid/structure coupling model. Also thermal losses in the boundary layer have not been considered.



(a) Results of absorption of a sound wave in a porous media exclusively computed with an LB flow solver.

(b) Typical absorption coefficients as a function of the frequency for a porous absorber with different layer thicknesses [52].

Figure 7.2.2 – Absorption coefficients

Future research on sound attenuation in porous media will follow two different approaches: First, the achievement of better results thanks to more advanced numerical experiments simulating e.g. a standing wave; second, the determination of material properties, such as permeability, tortuosity, viscous and thermal length [106], which can be used as input parameters for absorption models [104].

---

[1] To get results taking into account containing resonance effects, it would be more appropriate to simulate a standing wave. However, this has not been done, since the used experimental setup is still very demanding in terms of computational resources, but it is planned to perform such experiments in the near future.

8 Pre- and Postprocessing

While performing CFD simulations, a huge amount of data is created and even for complex geometries, needed as input data. These data have to be processed, archived and far more important during and after processing, transformed into meaningful display formats (e.g. charts, diagrams or animations). The steps necessary to process these data is divided in pre- and post-processing. Preprocessing is the essential step towards the simulation process executed by a numerical tool. During preprocessing a simulation domain is initialized by applying geometric boundary conditions in a way that the simulation kernel can process it. Typically, a digital model is transformed into a mesh type data structure, which has to correspond to the numerical model. Solvers based on lattice Boltzmann Methods typically use uniform or hierarchical block structured grids, which are briefly introduced in chapter 5. Post-processing is the method to examine and analyze the flow field solutions, including contours, vectors, streamlines, iso-surfaces, and animations to show the transient evolution of fluid flows. Post-processing becomes increasingly important. Section 8.2 describes the operation of common post-processing tools in serial and parallel, their capability in evaluating scientific data as well as their potential in visualizing CFD results. The last section 8.3 presents an approach, where CFD data is visualized using ray tracing techniques to simulate a more natural aspect instead of the very synthetic images commonly produced.

8.1 Preprocessing

The central part of the preprocessing is the grid generation already described in chapter 5. However, besides the actual creation of a grid, the process of decomposition and distribution of this grid is also important.

The simulation kernel comes with some intrinsic test cases, which can be used for e.g. validating model extensions or testing setups. In that condition, the whole grid generation process, decomposition, and distribution is managed by the kernel itself. If the numerical grid is created in advance, e.g. during simulations in natural porous media, the decomposition and distribution process has to be executed by external tools. The decomposition of the source domain is performed in a very simple way: because the initial grid is always a rectangular box, it is either decomposed uniformly as in Figure 8.1.1 or, as an alternative, the decomposition tool counts all fluid and boundary nodes and generates slices, containing the same amount of these active nodes at a certain axis (Figure 8.1.2).

This method for decomposing a complex geometry may sound very simple, even in terms of load balancing, but the nature of the mainly utilized porous materials mapped to uniform grids resulted in a surprisingly high parallel efficiency. This is due to the fact, that the pore space is evenly distributed and the porosity ranges between 30% and 60% (Figure 8.1.1).

For samples with a more heterogenous distribution of pore space, the slice algorithm has been

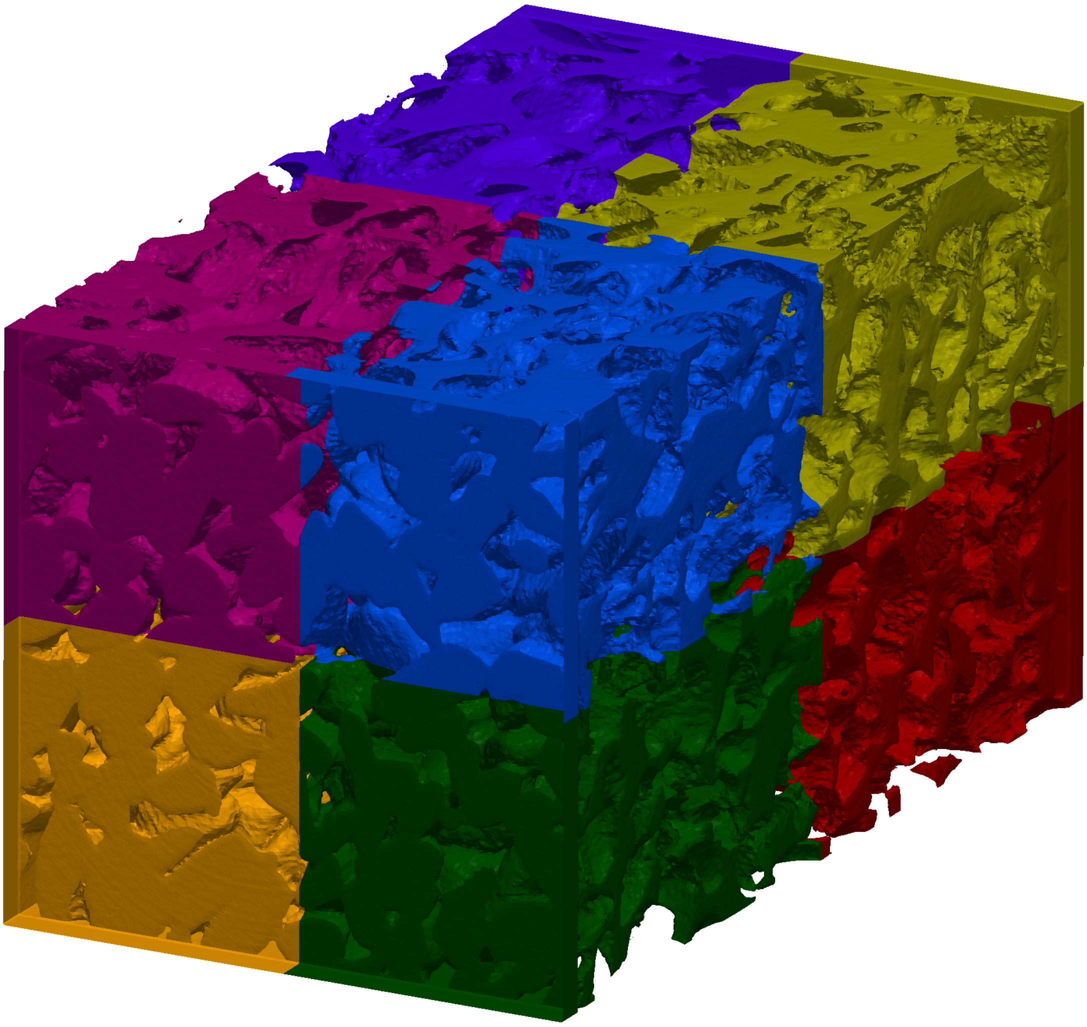


Figure 8.1.1 – uniform domain decomposition of a porous media

used (Figure 8.1.2). The drawback is a significant increase in communication time at the boundary of each sub domain during computation. In general, for porous materials with a far smaller porosity, other decomposition techniques may be more useful. For example multilevel recursive-bisection, multilevel k-way, and multi-constraint partitioning schemes such as implemented in METIS [82] or a recursive bipartite algorithms such as implemented in divide [63]. This last reference also shows comparisons between METIS, ParMETIS and divide in terms of resource requirements, performance and resulting efficiency. An implementation of a lattice Boltzmann flow solver using block type hierarchical grids and METIS as decomposition tool with all positive and negative aspects is described in [49].

Finally, a compromise has been made; the data structures of the lattice Boltzmann kernel are kept simple, as well as the decomposition process (and therefore fast, efficient, and more robust). In counterpart, one pays the unbalanced payload of the sub domains. However, this is mostly compensated, due to an uniform distribution of pore space, and can also be tuned with a careful manual choice of the decomposition topology.

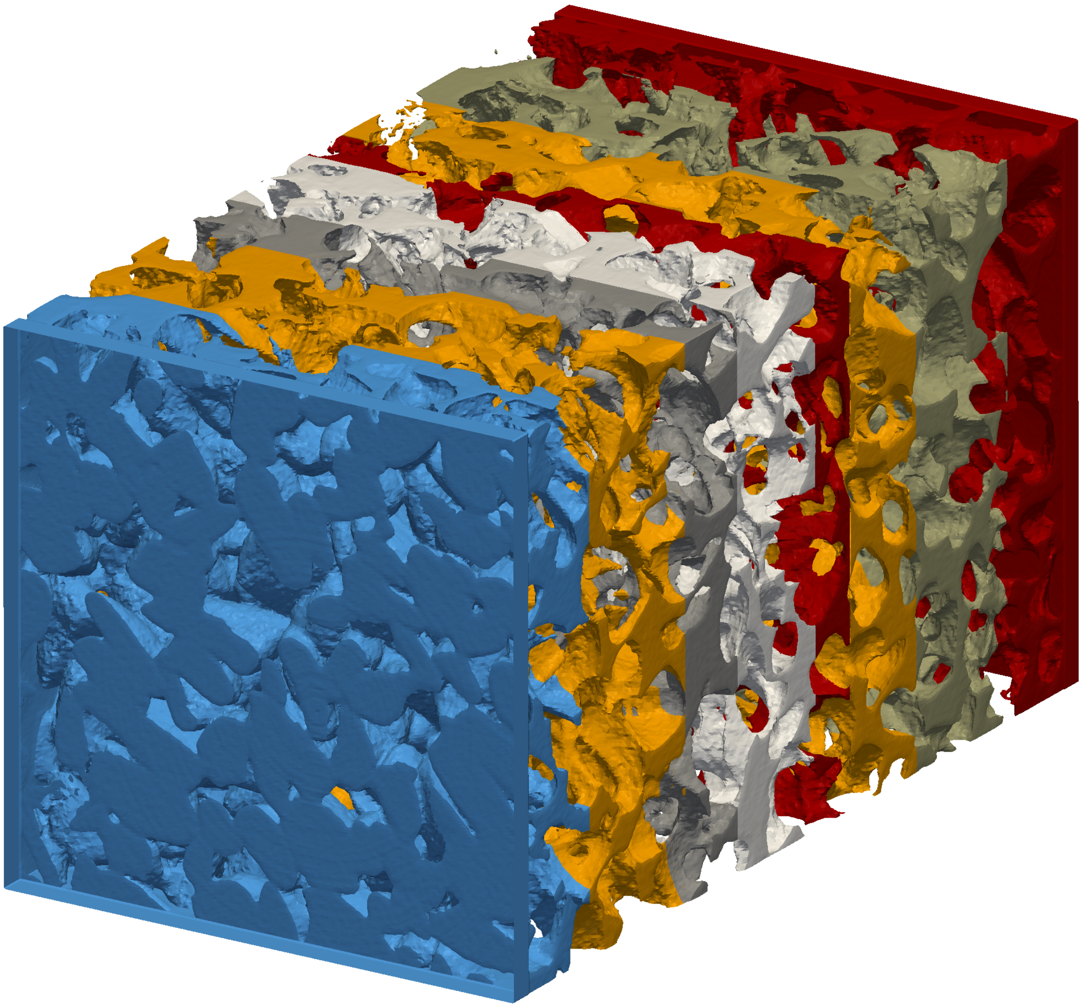


Figure 8.1.2 – domain decomposition with weighted slices

8.2 Post-processing

Post-processing is the final step in CFD analysis, it involves the organization and interpretation of the predicted flow data and the production of images and animations. During post-processing different variables have to be calculated in preparation for the analysis. Additionally, if parallel computation is used, MPI functions are required to combine data from different sub domains. Some of those variables are e.g.:

- derived variables: vorticity, shear stress
- integral variables: forces, lift/drag coefficients, mean velocity
- macroscopic quantities: minimum/maximum pressure or velocity, saturation
- turbulent quantities: Reynolds stresses, energy spectra

A variety of decisive quantities are naturally computed during runtime. They are needed to either predict the development of the simulation or as an abortion criterion. Besides the processing at runtime, the simulation kernel dumps data of the whole flow domain periodically to non-volatile memory, like a local hard disk or a network attached storage (NAS). This data contains the density, accordingly the pressure, the velocity components and depending on the type of simulation, the phase field, the temperature field, or the species concentration. To continue processing, the data has to be collected from all computational domains and combined into one dataset or processed directly in parallel.

Typically the dump files are combined into a single file using a simple post-processing tool from the VirtualFluids Classic framework. This is a standard procedure on most supercomputing machines, because the user has access to a specialized file system,[1] which provides access to all dumped data independently of the actual execution place within the machine. Yet, some HPC machines have a different architecture and the data is dumped locally on each computing node. The collecting process is then more difficult, because a specialized tool has to copy the data from each computational node to a separate file system. Afterwards the assembling of all sub domain dumps can be started. To accelerate this whole process a new tool has been developed which utilizes the MPI library to collect and assemble the data simultaneously. Also arbitrary downsizing functions have been integrated in order to collect a more compact/sparse set of the results, even during runtime. If the field data dumped by the kernel is collected and assembled during post-processing, the analysis of parts, or even the whole flow field, can be performed using scientific visualization tools or other techniques, like e.g. Fast Fourier Transform (FFT) or phase averaging.

Microprocessor performance (CPU, GPU or even specialized chip) has been improving roughly at a rate of 60% (Moore's law [109]) per year. Memory access times, however, have improved by less than 10% per year - see also Figure 8.2.1. This partly leads to the dilemma, that the time consumed for computing a CFD simulation is orders of magnitude lower than the time needed for preprocessing, transfering, and post-processing (including visualization) of the results. During CFD simulations, huge amounts of data are created, which nowadays ranges in terms of Gigabytes or even Terabytes. The data processing using conventional approaches is becoming increasingly difficult. Not only it is inevitable to parallelize the simulation kernel itself, but also the supporting tools dealing with pre- and post-processing have to be scheduled in parallel.

---

[1] A shared disk file system (also called shared storage file system, SAN file system or even cluster file system) is primarily used in a storage area network where all nodes directly access the block storage, where the file system is located.

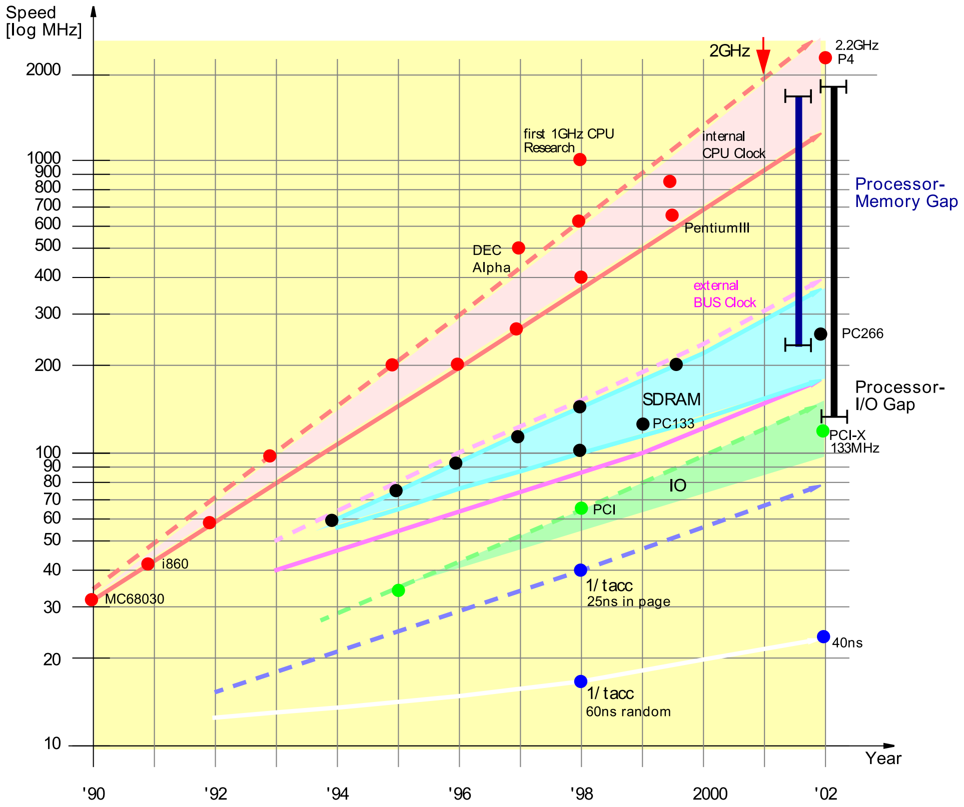


Figure 8.2.1 – Illustration of the development of arithmetic performance, main memory transfer rates, and latency

An alternative for collecting and preprocessing CFD data in a serial manner is the analysis through direct parallel post-processing. Thereby, the produced data are directly accessed by data servers on the computing nodes and then visualized by multiple render servers which are located on the computing nodes as well or separately on a render farm like a GPU cluster. A promising and capable tool for parallel post-processing is called ParaView [85], developed by Kitware, Sandia National Labs, and CSimSoft. ParaView is based on the Visualization ToolKit [126] and was developed to analyze extremely large datasets using distributed memory computing resources. It can be run on supercomputers to analyze datasets of terascale as well as on laptops for smaller data. Rapid development in the past years and its open source license have led to an extensive use of ParaView during the evolution of this thesis. Various setups (as illustrated in figures 8.2.2) have been used to utilize ParaView as a visualization tool of distributed data.

8.3 Visualization

> “A picture paints a thousand words”

It is obvious that one needs knowledge about the data and the environmental context to make an assumption about the meaning of the outcome of a simulation. Typically, during a CFD computation a lot of data are produced. These data are either evaluated by looking at the

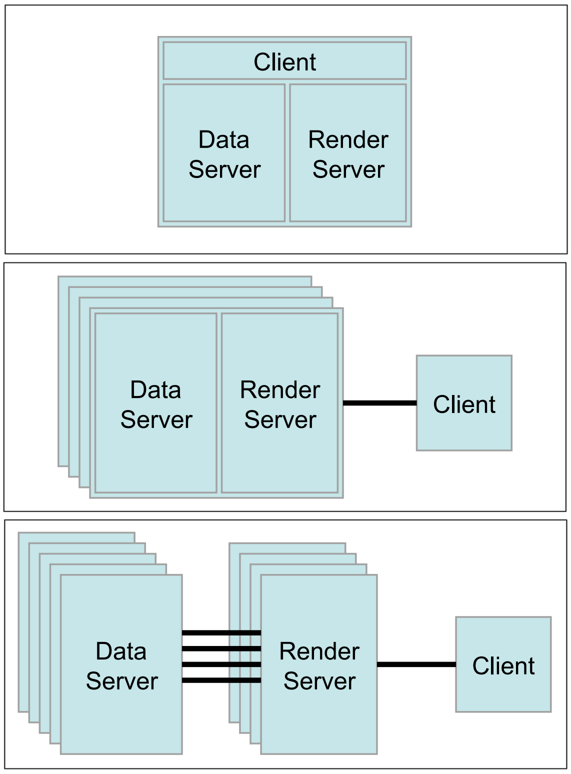


Figure 8.2.2 – Setup schemes for distributed data processing and rendering with ParaView

plain numbers, which may be useful in certain cases, or by using the whole repertoire of image generation to be efficient and as accurate as the predefined requirements (like e.g. animations as a tool to study time-dependent developments of vortical or turbulent structures and their interactions). During the development of this work and while working on different projects, the need to visualize scientific data was inevitable and even sometimes a challenge. This chapter describes typical visualization pipelines from raw transient data to a decent still image or even an animation of a flow process.

Scientific visualization has been formed into an independent discipline in 1987 were [102] published a report named “Visualization in Scientific Computing”. Since then tools have been developed which aid the natural scientist as well as the engineer by investigating the results of an experiment or an simulation not only by plain numbers but also as an synthetic image. In the natural and engineering sciences, three types of numerical variables gather almost all numerical data: scalars, vectors, and tensors. Typical examples for scalar variables are temperature, pressure, and density. Vector variables are velocity, vorticity, force, and any kind of scalar gradient like temperature gradient. Tensor variables describe for example stress, strain or the rate of deformation. There are a lot of distinguished visualization concepts for these kind of data - like coloring, height fields, contouring, or volume visualization, to pick just a few. A deep insight in visualization algorithms, data structures and API’s can be found in [126].

The simulation kernels used in this work are of non commercial nature and mostly scientific prototypes. Some special data, which can be calculated easily, are displayed during runtime, e.g. stability conditions, drag and lift forces, mean velocity, or saturation. However, the kernels do not have the capabilities to visualize flow data during runtime[2] or after completing a simulation task. Therefore additional tools are needed to evaluate the results. While section 8.2 describes the collection of the distributed result sets, a brief overview of scientific visualization software is given in the following.

The diversity of products capable of visualizing CFD data is limited and also split into commercial and non-commercial branches. On the commercial side tools like AVS[3] , Amira[4] , Ensight[5] , COVISE, FieldView, IRIS Explorer and Matlab, which are based on more or less proprietary frameworks and on the other side applications like ParaView, VisIt[6] , Vistrail, or OpenDX which are mainly based on the Visualization ToolKit[7] (vtk). All of the mentioned applications have its benefits and - unfortunately - drawbacks. During the creation of this work the following tools have been used for different purposes: Amira and Paraview mainly for extracting tomography data and converting it into various formats to be processed by the grid generation tools. AVS and ParaView (rarely Amira) for visualizing the data created by the simulation kernel. The visualization toolkit itself has been used as a library if none of the tools mentioned above could be successfully utilized.

One of the reasons for the effectiveness of scientific visualization may arise from the fact that 90% of all outside stimuli that a human can perceive is processed by vision. The visual representation is of great help in communicating the results of a simulation to others, scientist or not. Visualization also assists in the debugging process and in the development of physical and mathematical intuition. While numerical models, their implementation into software, and the performance of hardware are increasing rapidly, the demands in visualizing the (evolving) sim-

---

[2] At least one library exists, which can be linked to the kernel and is also capable of displaying visualizations of the flow data during runtime: the DISLIN [107] graphics library, developed by the Max Planck Institute for Solar System Research. DISLIN is written in Fortran and C. The name DISLIN is an abbreviation for Device-Independent Software LINdau since applications were designed to run on different computer systems without any changes. The library contains subroutines and functions for displaying data graphically, like curves, bar graphs, pie charts, 3-D color plots, surfaces, contours and maps. DISLIN interfaces C, Fortran, Perl, Python and Java. However, due to the massive parallel nature of VirtualFluids Classic and its distributed data concept, the management of data visualization through the kernel itself has not been further pursued.

[3] AVS/Express [4] (commercial), developed by Advanced Visual Systems, is based on a proprietary approach that utilizes a data flow architecture, consisting of different modules (functional units) and links between them to coordinate the data flow. AVS may be soonest comparable to Amira or VisTrails and is available for Windows, Linux (x86, x86-64 and IA64), Mac OS X, Solaris, HP-UX, IRIX and Alpha Tru64. AVS/Express is available in various editions, from single user up to parallel data processing and rendering.

[4] amira [156] (commercial), developed by Visage Imaging, Inc. is based on the Open Inventor toolkit [132]. Its main focus aims at medical imaging, however, it is also well suited for scientific visualization of large CFD datasets. Amira uses a modern user interface and a wide variety of data import functions, but is not capable yet of parallel execution and rendering.

[5] EnSight [26] (commercial), developed by Computational Engineering International, is like AVS also based on a proprietary toolkit and is available in different versions, from mobile visualization to cluster-based rendering. It is available for Windows, Linux (x86, x86-64 and IA64), Mac OS X, Solaris, HP-UX, IRIX and AIX.

[6] VisIt [41] (open source), developed by the Department of Energy (DOE) and the Advanced Simulation and Computing Initiative (ASCI) to visualize and analyze the results of terascale simulations. VisIt, is like ParaView, based on the Visualization ToolKit and employs a distributed and parallel architecture in order to handle extremely large data sets interactively. It is also freely available under a BSD license and operates on UNIX (Irix, Tru64, AIX, Linux, Solaris), MacOS X (10.3, 10.4), and Windows platforms.

[7] The Visualization ToolKit [126] (open source) is a very complete and powerful library for 3D scientific visualization. It is a software system for 3D computer graphics, image processing, and visualization used by thousands of researchers and developers around the world. VTK consists of a C++ class library, and several interpreted interface layers including Tcl/Tk, Java, and Python. However, it is not an end-user tool, it is a development package. A scripting language interface as well as a variety of C++ examples makes it possible to experiment with it rather quickly. Throughout this thesis some visualization examples have been directly processed with the help of vtk.

ulation results are increasing as well. In the past, results have been visualized using 2-D plots and tables. Nowadays it is common to use e.g. contour plots, volume rendering, animation, and even ray tracing techniques, as well as virtual reality tools. Although there has to be a compromise between scientific accurateness and aesthetics. In most cases during developing a plain visualization of synthetic still images or animations, as created by common visualization tools (ParaView, AVS, etc.), are perfectly sufficient. Sometimes, however, especially if the results are presented to an audience outside of the field of study, it may be helpful to beautify the transient or stationary results in terms of aesthetics. In the following, an introduction in using only open source tools to achieve visuals with the quality of a rendered CGI-movie is given.

8.3.1 Ray tracing in scientific visualization

The prerequisites are a set of software to simulate the required data, convert the results and finally render a scene. During this work the following software tools have been used:

- VirtualFluids and VirtualFluids Classic , for computing single phase, multiphase, thermal, and advection diffusion problems as described in Chapter 6 and briefly in Chapter 2.3.
- ParaView, a distributed rendering application used to either convert scalar field data into triangle meshes or convert files to different file formats such as the VRML successor X3D. A more detailed overview of the applications used to evaluate scientific data - also in parallel - can be found in the previous section.
- Blender, a lightweight and open source modeling and rendering tool which, runs on several popular computing platforms. Even if it is often distributed without documentation or extensive example scenes, the software contains features, that are characteristic of high-end modeling software. Blender has a robust feature set similar in scope and depth to other high-end 3D software, like e.g. Softimage|XSI, Cinema 4D, 3ds Max, Lightwave or Maya. These features include advanced simulation tools such as rigid bodies, fluid simulation, cloth and softbody dynamics, modifier based modeling tools, and character animation tools. The fluid simulation part used by Blender is incidentally based on a lattice Boltzmann model called El'Beem, which has been developed by [144]. More details can also be found in [143]. However, the LBM module in Blender is, due to its limitations, not used for simulating flow processes, but the infrastructure can be exploited to process the simulation results obtained with VirtualFluids Classic .
- A self written C++ application (based on vtk), to convert a time series of evolving fluid surfaces from a scalar (phase-)field into a triangle mesh. For this reason an iso-contour algorithm is utilized and, in a second step, the amount of triangles will be decimated using a decimate filter while preserving the topology of the surface. For the generated mesh, the quantity of triangles could be reduced by 10% to 90% without observing critical losses in quality. Finally the surface is saved into a propietary binary mesh format [143] used by Blender to store the timesteps of the included fluid simulation module.
- yafray, an alternative ray tracer, which in the contrary to the internal renderer of Blender supports caustics[8]. Unfortunately, development has been suspended. However, there are

---

[8] In an optical context, a caustic is an intensity pattern caused by reflections of light on a convex surface (catacaustic) or through diffraction in a transparent medium with an likewise convex surface (diacaustic). In 3D computer

lots of commercial and open source projects, that Blender can also export. For most of the popular rendering engines, 3rd party export scripts are available. Yafray has been used in some cases, where caustics are desired, e.g. when rendering multiphase simulations.

- yadra, a JAVA application for distributed frame based parallel rendering. Frame based rendering with Blender or similar rendering applications can be easily parallelized, because each frame can be rendered independently and out of order. By adding more computational resources, calculation times will decrease linearly. Also the distributed rendering of one single frame is possible (bucket rendering). However, the speedup in most cases is supposed to be sub linear, depending on the rendering methods (radiosity, ray tracing, ambient occlusion, environment maps, etc.) and because of the IO overhead. Nevertheless, most rendering applications make use of multi-core processors speeding up the development process. Yadra, in its present stage of development, only supports frame based rendering. During this work more than 12 Quad-Core PC-Workstations cumulating in 50+ CPU cores have been used to render animations of simulated fluid flow. The rendering times of an average animation with a couple of 100 frames decreased, from more than 8 days on a single core machine, to less than one hour when using distributed rendering. Fortunately the distribution approach chosen by yadra is completely parallel and on the contrary to other similar implementations there is no need for a central repository where all workstations have to have access to. Also, due to the fact that the implementation is written in JAVA, heterogeneous setups are possible, as Blender runs on all major platforms.
- JAVA Service Wrapper, which can install JAVA applications as an NT-Service: In order to have the most flexibility accessing workstations, which are only used during office- or class hours, it is suitable to install parallel render capabilities as a service/daemon, which runs without user interaction, is started automatically during boot time and can possibly run with a priority below normal. Thanks to this approach, regular user work is not affected and rendering capabilities are in place.
- VirtualDub is a video processing application, which is excellently suited for combining single frames into video streams of various formats. VirtualDub includes many filters for converting and processing video stream as well as an SDK for developing your own filters.

Combining all tools described above permits the visualization of numerically obtained data not only in a synthetic manner, but also in a more natural way up to a photo-realistic presentation. Nevertheless, the setup for a single scene is still very time consuming and therefore only occasionally acceptable. This case study and the appearance of e.g. POV exporters in more and more scientific visualization packages, however, shows that the demand for photo realistic presentation of scientific data is increasing.

---

graphics, caustics have been the most difficult part to simulate, because their appearance is concentrated and they can only be simulated through methods, which are guided by light sources. The photon mapping algorithm introduced by [79] was the first approach to simulate caustics generically and efficiently.

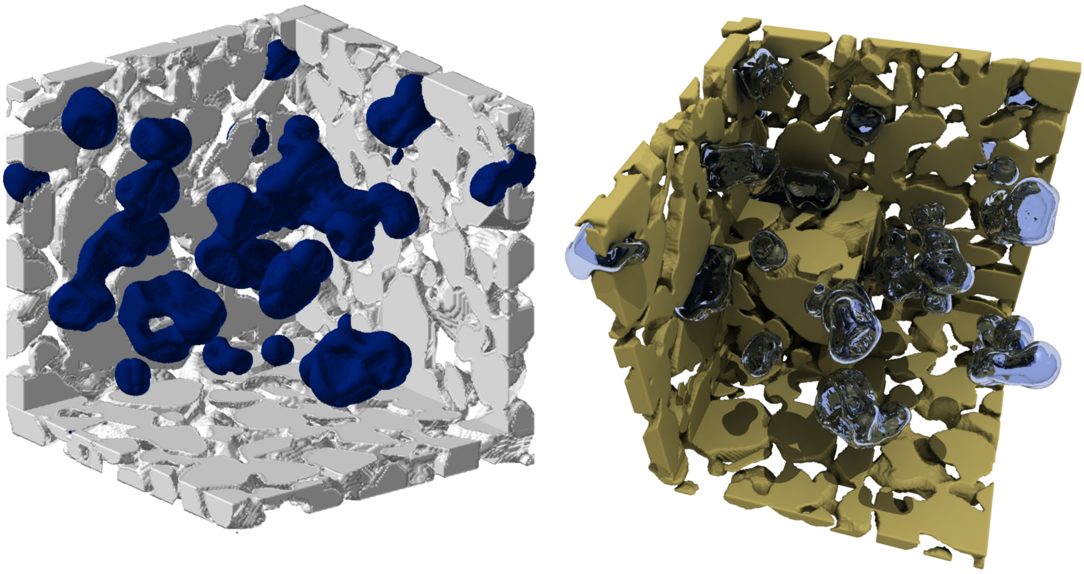


Figure 8.3.1 – Comparison of a regular visualization done by AVS versus one done with Blender

9 Implementation

A lattice Boltzmann simulation code can be implemented easily and almost independently of the utilized programming language. However, if boundary conditions become more complex, multiphysics/multicomponent models are introduced, the kernel has to be parallelized or even if adaptive- and non-uniform approaches are needed - the developer has to be very careful in considering design aspects and implementation techniques.
The following section describes the implementation and the design of the LBM simulation package used throughout the experiments. This package, namely VirtualFluids , consists of several branches: the one mainly used throughout this thesis, i.e. for multiphase flow in porous media, advection/diffusion and thermal problems, and acoustics. It has been written from scratch and in the Fortran90/2003 programming language (in the following referred to as VirtualFluids Classic or VFC). This branch is based on uniform grids and follows an MPI parallelization approach described in section 9.1. In various experiments other branches of VirtualFluids have been used as well, primarily a C++ implementation of a block structured data layout with grid refinement [49]. Furthermore the VirtualFluids framework (developed at the Institute of Computational Modeling in Civil Engineering (iRMB) at the University of Braunschweig) has also branches that are supporting fluid-structure interactions as well as free surface flow, turbulence, and thermal models [50, 53, 55, 78, 136, 145, 146].

9.1 Parallelization

The lattice Boltzmann method is typically resource intensive when it comes to larger 3D arrays, however, if the memory consumption is counted per lattice node the performance/memory ratio is excellent, even if compared to other state-of-the-art methods. Furthermore, if the geometry become complex, the memory demand stays constant. The lattice Boltzmann method generally requires only the information of the next neighbor, at least if the fluid wall interface does not use higher order interpolation schemes and the discretization stencil is locally available (D2Q17, D3Q27, and D3Q39 [131] stencils are non local and require special treatment if processed in parallel). Therefore the algorithm can be parallelized very easily by using communication library routines such as the message passing interface (MPI) library [124] or alternatives.
Due to the constraints given by the FIMOTUM-project agenda, [1] the main goals during implementation were to aim at best scalability, low memory consumption, high performance for the single- as wells as for the multiphase models, and the possibility to run simulations in very large domains, meaning e.g. natural porous media with a degree of freedom (DoF) of more than 2×10^{10}. During performance measurements simulations have been carried out with up to 4000^3 grid nodes, resulting in 1.280.000.000.000 DoF. The simulation kernel has been designed to be truly parallel meaning that no master node supervising the slave processes or distributing the

---

[1] The FIMOTUM project (**FI**rst Principle Based **MO**delling of **T**ransport in **U**nsaturated **M**edia) has been granted under KR 1747/7-2 by the German Science Foundation (DFG) and was the framework for most of the contributions to this thesis.

geometry is needed. Also the kernel is targeted towards porous media simulations which, at least if the resolutions are large enough, does not require load balancing. Performance results for different test cases and architectures will be given in the section 9.2.
During the development of applications for distributed memory machines, the Message Passing Interface (MPI) is the approach most widely used. However, certain implementations of MPI are also suitable for shared memory architectures. The data which has to be exchanged at the simulation boundaries of each sub domain will be transfered via Random Access Memory (RAM) if the MPI processes share the same address space. This is commonly the case in shared memory architectures [62]. A further advantage of MPI is the automatic selection of a dedicated network if processes have to communicate across shared memory. In contrast to the parallelization technique OpenMP [39] which works exclusively on shared memory systems, MPI is far more flexible. The drawback here is that MPI requires more low level work from a developer and an upfront design while OpenMP directives can often be implemented afterwards into a serial working code. Also, the simulation domain has to be correctly split into packages that each node can process. For this purpose the preprocessing tool described in section 8.1 has been developed.

9.2 Performance

> "I do not fear computers. I fear the lack of them."
> Isaac Asimov

The performance of a numerical code depends on the algorithms used, the kind of implementation, the data structures and the execution speed of the hardware. Unfortunately, the performance of the hardware depends not only on a single CPU's peak performance, it is also constraint by the available memory bandwidth (see also section 8.2 and Figure 8.2.1 within). A more detailed discussion on core performance, memory bandwidth and favorable data layout is given in [160]. For multiprocessor systems using a distributed memory approach, such as MPI, the network bandwidth and the network latency have also to be considered. The performance P of an LB implementation is measured in **l**attice **u**pdates **p**er **s**econds (LUPS), meaning how many lattice nodes can be updated in one second. The collision and the propagation part are split into two steps (iterations), during collision all arithmetic operations are performed and the propagation streams the values to the neighboring nodes (and therefore accesses the main memory). A rough estimation for the attainable maximum performance of the LB algorithm for single phase problems on one CPU core in LUPS is then given by [149]

$$\mathsf{P} = \min\left\{\min\left\{\frac{\mathsf{MBW}}{2\,\mathsf{NB}}, \frac{\mathsf{PEAK}}{4\,\mathsf{NF}} + \frac{\mathsf{MBW}}{4\,\mathsf{NB}}\right\}, \frac{\mathsf{MBWNet}}{\mathsf{NBEx}}\mathsf{RAT}\right\} \tag{9.2.1}$$

where MBW is the maximum bandwidth to memory interface, NB is the number of bytes per cell to be transferred from or towards the main memory per iteration, PEAK is the peak performance, NF are the number of floating point operations per cell for a collision operation, MBWNet is the network bandwidth and NBEx is the number of bytes per communication node to be exchanged between two cores. RAT is the smallest ratio of inner plus outer (communication) nodes over outer nodes on a single core. Considering the memory bandwidth as the limiting factor and assuming a write allocate strategy for scalar CPU architectures (an additional cache line load is performed on a write miss), one finds $\mathsf{NB} = (\ 20\ \text{(read)} + 2 \times 19\ \text{(write)}\ ) \times 4$ bytes $= 232$ bytes per cell for the D3Q19 model. While utilizing the Intel Itanium2 Montecito Dual Core

(1.6 GHz), the bandwidth has to be shared between two cores and a bandwidth MBW of 4.25 Gigabyte/sec(GB/s) is obtained. While memory bandwidth is limited by the architecture, the average number of floating point operations NF per cell depends on the processor properties, such as the number of FPUs, etc. as well as on the compiler optimization strategies.
The D3Q19 MRT (single phase) algorithm has to execute approximately 220 additions and 60 multiplications per lattice cell. Since the peak performance can only be achieved if the processor's core can perform an addition and a multiplication simultaneously $\mathsf{NF} = 2 \times 60 + 2 \times (220 - 60) = 440$ is chosen. Exemplarily, the Intel Itanium2 Montecito Dual Core with 1.6 GHz internal clock speed has a performance of PEAK=6.4GB/s. To compute the maximum theoretical value of P_{max} for a single core, one has to choose the minimum value for the collision and the propagation restricted by bandwidth and peak performance. This is reflected by Equation 9.2.1 and the resulting value is $P_{max} = 8.1E6\mathsf{LUPS}$.
The data which has to be exchanged over the network depends on the simulation setup. For a single block the kernel has to exchange 6 planes with 5 mass fractions and 18 lines with 19 mass fractions. The number of bytes NBEx, which have to be transferred, depends on the size and the extensions of the blocks. While most simulations have been performed on a supercomputer of type SGI Altix 4700[2] this model is considered as the reference frame for performance measurements. The bandwidth of the network of this model depends mainly on the number of cores utilized. It is assumed that one network connection is shared between 2 cores and that, due to the partitioning of the Altix 4700 system, large jobs have only 0.1 or 0.2 GB/s of network bandwidth. Therefore the following bandwidth/process ratio applies:

$$\mathsf{MBWNet} = \begin{cases} 3.2GB/s & \text{if numProc} <= 8 \\ 0.8GB/s & \text{if numProc} <= 510 \\ 0.4GB/s & \text{if numProc} <= 1020 \\ 0.1/0.2\,GB/s & \text{else.} \end{cases} \tag{9.2.2}$$

To analyze the performance of the LB kernel, different resolutions of grid sizes of 240^3, 1000^3, 2000^3 and 4000^3 have been investigated. In this section it has been assumed that the maximum achievable single core performance - due to limitations in the bandwidth to the memory interface - can be estimated with P=8.1E6 LUPS. For single phase simulations a value of P=3.33E6 LUPS (see Table 9.2.1) is obtained and corresponds to 41 % of P_{max}.

To estimate the limitation in performance P by the network bandwidth, the amount of data which has to be exchanged for each process has been computed. Also an average value NBEx (in byte) for each lattice node participating on the communication has been determined. Assuming a cube of n^3 grid points, a number of $nproc$ cores and a simple domain decomposition one can obtain a side length of the subdomains related to one core $nloc = (n^3/nproc)^{(1/3)}$. The number of nodes $NCloc$ participating on the communication can be estimated by $NCloc = 6\,(nloc - 1)^2 + 12\,nloc$ and the number of bytes to be transferred in one direction by $ByteAll = 6 \times 5 \times 4byte \times (nloc - 1)^2 + 12 \times 19 \times 4byte \times nloc$. The amount of bytes to be transferred per

---

[2] Computations have been performed on supercomputing infrastructures at the LRZ in Munich (SGI Altix 4700) [149], the SCC in Barcelona (Mare Nostrum - PowerPC architecture), and at the iRMB in Braunschweig (AMD Opteron Cluster).

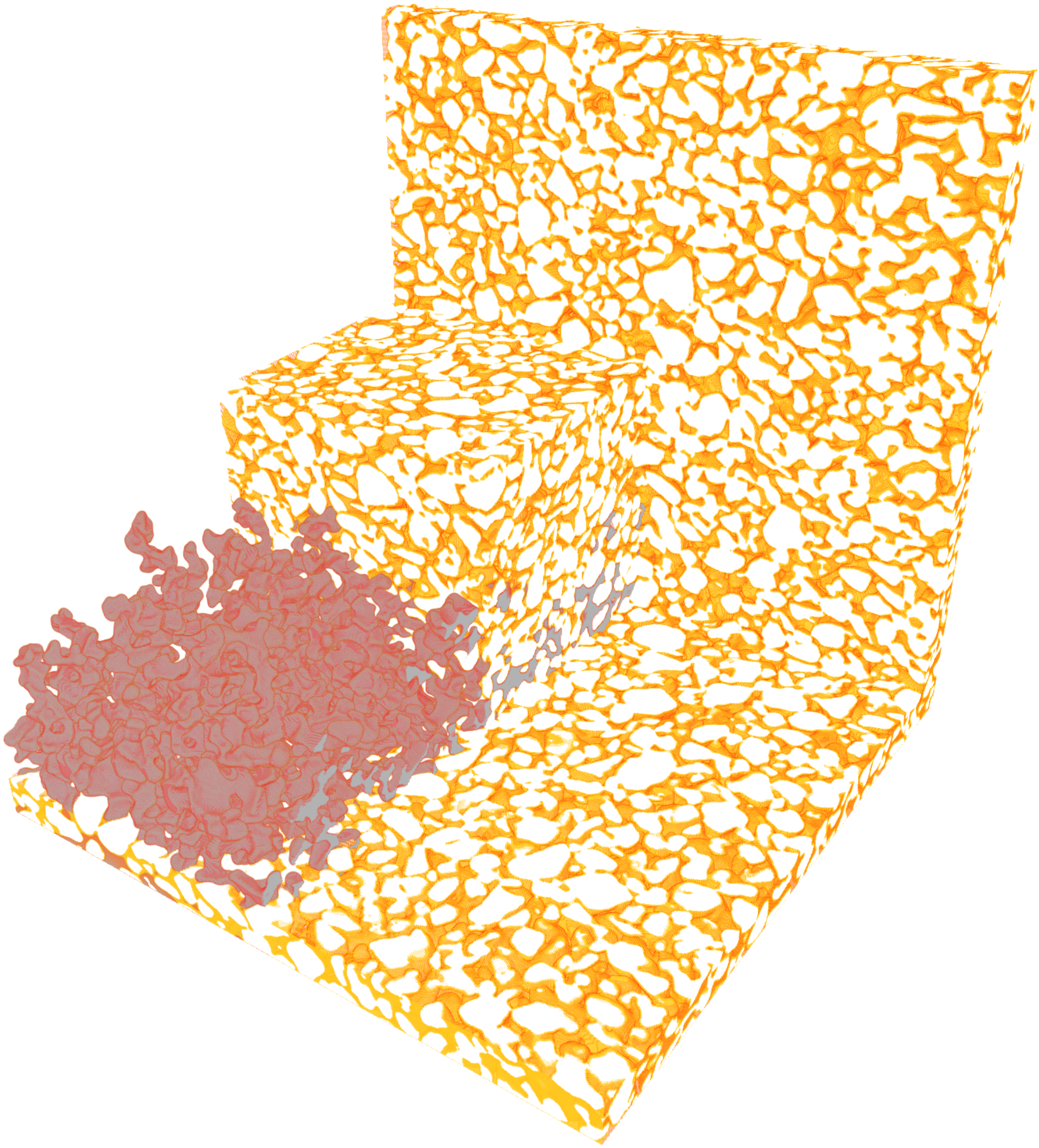


Figure 9.2.1 – Setup for determine performance in multiphase systems: drainage simulation in porous media at full resolution (800^3 grid nodes)

communication node per time step can be estimated by $\mathsf{NBEx} = 2\,ByteAll/NCloc$. Finally, the ratio RAT given in equation 9.2.1 can be computed by $\mathsf{RAT} = nloc^3/NCloc$. The restriction in LUPS by the maximum bandwidth of the network is then given by

$$R_{max,net} = \frac{\mathsf{MBWNet}}{\mathsf{NBEx}}\mathsf{RAT} = \mathsf{MBWNet}\frac{nloc^3}{240(nloc^2-1)+1824\,nloc}. \tag{9.2.3}$$

In Table 9.2.1 the Performance P, the bandwidth of the network BW_{net}, the fraction of the communication f_{net}, the restriction $R_{max,net}$ by the maximum bandwidth of the network and the efficiency for different problem sizes are given. Even for a relatively small grid size of 240^3 a very good scaling behavior is achieved. As long as $R_{max,net}$ is larger than the theoretical single core performance (8.1E6 LUPS), the efficiency is still very good. Even for a number of more than 1000 cores a super-linear scaling is observed - due to caching effects. For 2028 cores $R_{max,net} = 6.09E6$ LUPS is smaller than the single core performance and the parallel efficiency degrades. The latency itself is insignificant, because even for the smallest sub domains the time needed for one time step is much larger than the latency itself. For larger domains this behavior can also be observed, however, it is not clearly pronounced. In general a very good scaling behavior of the code is observed, even when using 4080 cores for a grid of $\mathbf{4000^3}$ parallel efficiency of **92 %** is obtained. Finally, in Figure 9.2.2 the parallel efficiency versus the number of cores for different grid sizes are summarized.

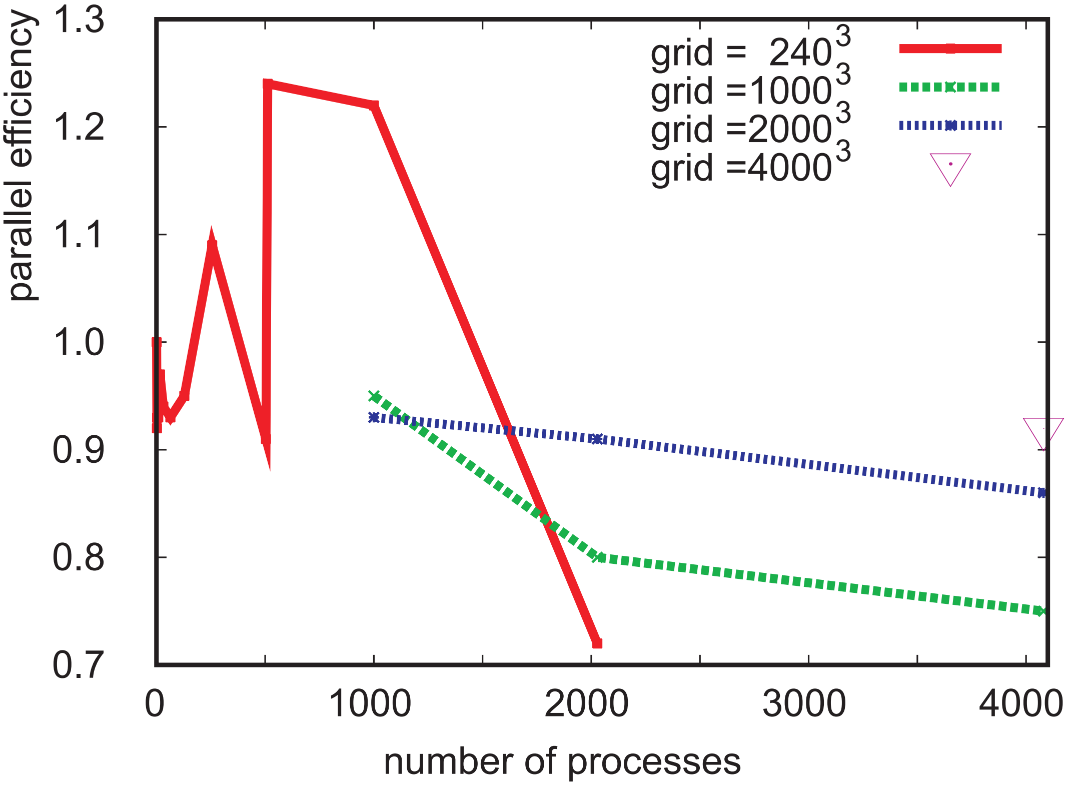


Figure 9.2.2 – Parallel efficiency of different problem sizes

Table 9.2.1 – Performance P, the bandwidth of the network BW_{net}, the fraction of the communication f_{net}, restriction $R_{max,net}$ by the maximum bandwidth of the network and efficiency $eff.$ for different problem sizes

# cores	P [LUPS]	BW_{net} [GB/s]	f_{net} [-]	$R_{max,net}$ [LUPS]	$eff.$ [-]
grid: 240^3					
1	3.33 E6	-	-	-	-
2	6.16 E6	3.2 E9	0.02	2.47 E9	0.92
4	1.24 E7	3.2 E9	0.04	1.94 E9	0.93
8	2.49 E7	3.2 E9	0.03	1.53 E9	0.93
16	5.16 E7	8.0 E8	0.05	3.00 E8	0.97
32	1.01 E8	8.0 E8	0.06	2.35 E8	0.94
64	1.95 E8	8.0 E8	0.11	1.83 E8	0.93
128	4.05 E8	8.0 E8	0.09	1.42 E8	0.95
256	9.32 E8	8.0 E8	0.19	1.10 E8	1.09
504	1.52 E9	8.0 E8	0.42	8.47 E7	0.91
512	2.12 E9	4.0 E8	0.19	4.21 E7	1.24
1000	4.05 E9	4.0 E8	0.25	3.23 E7	1.22
2028	4.88 E9	2.0 E8	0.57	**6.09 E6**	0.72
grid: 1000^3					
1000	3.16 E9	4.0 E8	0.06	1.58 E8	0.95
2028	5.41 E9	2.0 E8	0.25	3.07 E7	0.80
4080	1.02 E10	2.0 E8	0.29	2.39 E7	0.75
grid: 2000^3					
1000	3.10 E9	4.0 E8	0.04	3.24 E8	0.93
2028	6.16 E9	2.0 E8	0.07	6.36 E7	0.91
4080	1.17 E10	2.0 E8	0.11	4.99 E7	0.86
grid: 4000^3					
4080	1.25 E10	2.0 E8	0.07	1.02 E8	0.92

10 Conclusions

The main objective of this thesis was to develop and implement a properly designed tool for CFD computations based on the lattice Boltzmann method, which is capable in running efficient simulations of multiphase flow in natural porous media at a large computational scale ($2E11+$ DoF). The following section will summarize the contributions of this thesis that were presented in order to achieve those objectives. This includes the description of the method, an extensive validation part, followed by experiments to determine the material properties of multi phase flow in porous media and by issues regarding the processing of CFD data. This chapter is concluded by a discussion of drawbacks and future extensions.

10.1 Summary

This Thesis has presented an efficient numerical tool to solve single and multiphase flow especially designed for large scale computations and complex geometries. Also results for single and multiphase flow have been presented and compared to an alternative model as well as to experimental findings.

The first part (Chapter 2-4) addresses the description of the numerical method, the boundary conditions, and the validation examples to verify a consistent implementation. The most relevant conclusions and novel contributions for this first part are:

- In contrary to LBGK models with Single-Relaxation-Time utilizing the simple bounce back scheme, the MRT model in combination with higher order boundary conditions are more accurate. Therefore, and due to the grid decoupling approach, coarser grid resolution can be chosen as for LBGK type simulations of the same geometry would be necessary.The additional costs in terms of preprocessing and decreased locality of the boundary stencil are more than balanced by the increase in numerical efficiency.
- Various validation simulations have been performed (e.g. permeability, pendular rings, capillary rise, or residual water saturation). The results have been compared to other modeling approaches and to experimental data; they show in consideration of the complexity of the problems a very good agreement.
- Due to the changes and additions made to the lattice Boltzmann simulation kernel the parameter space could be greatly enhanced. The Multiple-Relaxation-Time model and the separate advection of the phase field are the core enhancements for the multiphase simulations. That made it possible to perform simulations in porous media at a large scale, as well as the determination of material functions, which have not been obtained in that accuracy before.
- Besides no-slip boundary conditions of higher order, a free-slip type has been introduced to enable the measurement of relative permeabilities in single phase contexts. This permits the determination of relative permeabilities without solving the whole multiphase problem.

- The MRT type lattice Boltzmann model has also been extended in terms of advection/diffusion capabilities, which permits the computation of transport problems in porous media. Furthermore a thermal extension, based on a finite difference scheme, has been added. Therefore problems involving thermal flows can be computed.

In the second part of the Thesis (Chapters 5-7) the grid decoupling process is introduced, which allows the efficient execution of various flow experiments in natural porous media. The main conclusions are:

- Applying the grid decoupling approach demonstrates that second-order accuracy and a significant speed-up in convergence can be achieved not only for flow simulations based on CAD-type geometries, but also from scanned data such as porous media flow in sand. Due to the decoupling, convergence studies in natural porous media can easily be performed and could raise the computational efficiency by one order of magnitude [5].
- Only a few studies [1, 2, 11–13, 101, 150, 164] have reported simulations of multiphase flow in three-dimensional porous medium systems, in part because of the computational limitations. Due to the enhanced lattice Boltzmann framework and the high quality scans of natural sands extensive studies of single- and multiphase flow in porous media have been carried out during the project duration. New approaches in determining the constitutive relationships have been developed [7].
- The capability of two different multiphase models (LBM & MPNM) to predict the hysteretic behavior of the capillary pressure saturation relationship has been evaluated. This has been done on the basis of detailed representations of the complex porous structure as measured by X-ray or Neutron tomography. The models differ considerably in complexity and computational costs and can have different potential applications. However, the results obtained for the primary and secondary drainage curve are consistent, while differences among the models for the first imbibition are observed.
- Only with the LB approach it is possible to simulate a fully resolved simulation of dynamic multiphase phenomena tracking the interface evolution. The air entry pressure and the slope of the drainage curves can be predicted by the LB very well. However, the standard LB approach using uniform grids is limited in the dry range, due to the insufficient resolution of thin water films.

In the last part of the Thesis (Chapters 8-9) issues of large scale computing are addressed. Scientific visualization as a tool for analyzing CFD data is a subject of discussion. This part is then concluded by notes about implementation and measurement results regarding parallel efficienca, scalability and performance. The most relevant conclusions are:

- Due to the increasing challenge in processing very large CFD datasets, parallel data processing has been applied using proprietary tools developed within the framework of this thesis as well as tools dedicated to scientific visualization such as ParaView or VisIt. Furthermore, a toolchain has been created towards photo realistic visualization of simulation results utilizing ray tracing techniques (Blender) and parallel rendering capabilities (Render farm).
- In general the properly designed implementation of the LB method has a very good parallel efficiency even for a large number of cores. For flow in porous media a good parallel efficiency is observed, even in consideration of the complexity of the problem. Also the speedup for medium sized problems with many cores is very good. Due to the Fortran95/2003 programming language the simulation kernel can easily be adapted to various

super computing architectures. This has been done e.g. for the cluster at the iRMB (120 CPU Opteron Linux-Cluster/MPI), to the PowerPC/BlueGene architecture (MareNostrum at the Super Computing Centre in Barcelona) or SGI/Altix Systems (HLRBII, Munich).

10.2 Outlook and Future Work

In general the lattice Boltzmann method proved to be an efficient technique for detailed investigations of dynamic multiphase phenomena. However, there are disadvantages of the method which have to be substantially reduced in order to simulate physical correct flow processes especially in porous media. Furthermore, not only the numerical model has deficiencies, but also during pre- and post-processing improvements could be made. The most pressing issues are:

a The computational costs are very high, even while using decoupled grids, however this is a typical problem in the field of CFD simulations, even for alternative flow solvers.

b The standard LB approach used in this thesis is based on uniform grids and has therefore limitations in the dry range (which is typical for every CFD method using uniform grids). This is caused by an insufficient resolution of thin water films and thus leads to erroneous capillary pressures.

c Large density ratios are difficult to obtain in the presented multiphase model. The stress tensor is modified in an non-physical way, leading to a disturbance at the fluid/fluid interface. However, recent progress in lattice Boltzmann modeling using level set methods shows that these drawbacks can be resolved [142].

d More generally, a comparison of simulation results and physical experiments is flawed by the fact that certain effects can influence the macroscopic measurements (compaction at top and bottom, large pores at the container wall) that are not reproduced in a very small scanned sample.

In order to model film flow in a physical correct (b) way with reasonable computational effort (a), one would think about using grid refinement. Methods for simulations of two-phase flows using non-uniform grids have been developed e.g. in [146] and at present algorithms are developed where an adaptive block-wise strategy is followed. Here cuboids containing a fixed number of nodes which are the leaves of an hierarchical octree data structure and are dynamically (de-)allocated if needed [50]. Nevertheless, using grid refinement for single or even two-phase flow in porous media is a challenge, due to the surface/volume ratio.

The density ratio (c) of liquid-gas systems is usually larger than 100, and even the density ratio of water to air is about 1000. However, the mobility criteria in the experiments is usually met and therefore representing the physical behavior of air/water systems. Nevertheless, some efforts have already been made to remove the issues of a modified stress tensor, among other approaches, by using correction terms. These terms are substantially reducing the observed effect and even cancel it out in the asymptotic limit.

Uncertainties in the macroscopic measurements (d) of real world experiments could be avoided by measuring the fluid distribution in a small sample using X-rays from synchrotrons as described in [37]. Then a comparison between numerical models and experimental results could be far more meaningful. In the framework of this thesis no such data were available, however, experiments are in preparation supposed to deliver X-ray images of fluid distribution in porous media.

Future research, which is based on the developments in this Thesis, will aim on the simulation of relative permeabilities in porous media without solving the full multiphase problem utilizing the free slip boundary conditions as a substitute for a fluid/fluid interface. The gained results should be compared against measurements already taken during the scanning process.
The LB-method is also very well suited for acoustics problems. The application field which is currently under investigation is the determination of sound absorption properties of road surfaces. The geometry is again obtained by tomography methods. Since the number of time steps needed for an acoustics problem is much smaller than for a hydrodynamic simulation we can run simulations on very large grids and thus expect to be able to solve complex real life problems with sufficient accuracy.
Recently implemented model extensions have permitted the simulation of turbulent convective flows. Using a Smagorinsky subgrid scale model for fluid flow and heat flux and a finite difference scheme for solving the heat equation. Targeted applications are the convection inside buildings and air condition systems at realistic Reynolds and Rayleigh numbers.

The work carried out in this project formed the basis for productive scientific research. The methods developed will be used in the following planned international research project and DFG proposal:

- A proposal for an international research unit ‘Multi-Scale Interfaces in Unsaturated Soil’ is being funded by the DFG. The aim of this project is to combine measurement and modeling techniques in order to improve the quantitative understanding of dynamics of fluid interfaces in soils, focusing on the transition across soil-atmosphere interfaces as well as internal interfaces between solids and fluids. The interaction between material interfaces of different porous media and the stability properties of infiltration and evaporation fronts that decide flow paths and thus flow dynamics in heterogeneous structures gain special attention in the planned project.

A Transformation matrix M

$$\left[\begin{array}{lrrrrrrrrrrrrrrrrrrr}
1\cdot & (1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1) \\
c^2\cdot & (-1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1) \\
c^4\cdot & (1 & -2 & -2 & -2 & -2 & -2 & -2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1) \\
c\cdot & (0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0) \\
c^3\cdot & (0 & -2 & 2 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0) \\
c\cdot & (0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1) \\
c^3\cdot & (0 & 0 & 0 & -2 & 2 & 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1) \\
c\cdot & (0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1) \\
c^3\cdot & (0 & 0 & 0 & 0 & 0 & -2 & 2 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1) \\
c^2\cdot & (0 & 2 & 2 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -2 & -2 & -2 & -2) \\
c^4\cdot & (0 & -2 & -2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -2 & -2 & -2 & -2) \\
c^2\cdot & (0 & 0 & 0 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0) \\
c^4\cdot & (0 & 0 & 0 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0) \\
c^2\cdot & (0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0) \\
c^2\cdot & (0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1) \\
c^2\cdot & (0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0) \\
c^3\cdot & (0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 0 & 0 & 0 & 0) \\
c^3\cdot & (0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1) \\
c^3\cdot & (0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1)
\end{array}\right]$$

B Curriculum Vitae

General Information

Date / Place of Birth: 1975-01-20, Rostock, Germany
Office Address: Universität Braunschweig
Institute of Computational Modeling in Civil Engineering
Pockelsstr. 3, 38106 Braunschweig, Germany
Phone: +49 (0)531 391-7584
Email: ahrenholz@irmb.tu-bs.de
www: http://www.irmb.tu-bs.de

Education

Sep/2009 - present	Post-Doc at the Institute of Computational Modeling in Civil Engineering, Universität Braunschweig, Germany
May/2004 - Aug/2009	PhD-Student at the Institute of Computational Modeling in Civil Engineering, Universität Braunschweig, Germany
Dec/1999 - Oct/2000	Study of Civil Engineering at the Ohio State University, USA
Jun/2000 - Sep/2000	Research assistant at SeaGrant-Program “Automated In-Situ Profiling in the Bottom Boundary Layer”, Lake Erie/OH, USA
Oct/1995 - Oct/2003	Study and Diploma degree in Civil Engineering at the University of Hanover, Germany
Aug/1991 - Jul/1994	High School: Niedersächsisches Internatsgymnasium, Bad Harzburg, Germany. Degree: Graduation (Abitur)

Bibliography

[1] Adler, P., and Thovert, F. Real porous media: Local geometry and macroscopic properties. *Applied Mechanics Reviews 51*, 9 (1998), 537–585.

[2] Adler, P., Thovert, J.-F., Bekri, S., and Yousefian, F. Real Porous Media: Local Geometry and Transports. *Journal of Engineering Mechanics (ASCE) 128*, 8 (2002), 829–839.

[3] Adler, P. M. *Porous media: geometry and transports.* Butterworth-Heinemann, 1992. ISBN 0-7506-9236-7.

[4] Advanced Visual Systems, Inc. AVS/Express developer edition. Website, 2008. `http://www.avs.com`; accessed on December 30th 2008.

[5] Ahrenholz, B., Tölke, J., and Krafczyk, M. Lattice Boltzmann simulations in reconstructed parametrized porous media. *International Journal of Computational Fluid Dynamics 20* (2006), 369–377(9).

[6] Ahrenholz, B., Tölke, J., and Krafczyk, M. Second order accurate slip boundary conditions for arbitrary geometries in lattice-boltzmann simulations. Talk at DSFD 2006, 08 2006. www.irmb.tu-bs.de/.../DSDF06.pdf.

[7] Ahrenholz, B., Tölke, J., Lehmann, P., Peters, A., Kaestner, A., Krafczyk, M., and Durner, W. Prediction of capillary hysteresis in a porous material using lattice Boltzmann methods and comparison to experimental data and a morphological pore network model. *Advances in Water Resources 31*, 9 (2008), 1151–1173.

[8] Alexander, F., Chen, S., and Sterling, J. Lattice Boltzmann thermohydrodynamics. *Physical Review E 47* (1993), R2249.

[9] Ansumali, S., Karlin, I., and Ottinger, H. Minimal entropic kinetic models for hydrodynamics. *Europhysics Letters 63*, 6 (2003), 798–804.

[10] Autodesk. Autodesk Architectural Desktop. Website, 2006. `http://www.autodesk.com`; accessed on December 14th 2008.

[11] Bekri, S., and Adler, P. M. Dispersion in multiphase flow through porous media. *International Journal of Multiphase Flow 28* (2002), 665–697.

[12] Bekri, S., Howard, J., Muller, J., and Adler, P. M. Electrical resistivity index in multiphase flow through porous media. *Transport in Porous Media 51*, 1 (2003), 41–65.

[13] Bekri, S., Vizikab, O., Thovert, J.-F., and Adler, P. Binary two-phase flow with phase change in porous media. *International Journal of Multiphase Flow 27* (2001), 477–526.

[14] Bentley Systems, Inc. Microstation. Website, 2006. http://www.bentley.com; accessed on December 14th 2008.

[15] Benzi, R., Succi, S., and Vergassola, M. The lattice Boltzmann equation: theory and applications. *Physics Reports 222(3)* (1992), 147–197.

[16] Bhatnagar, P. L., Gross, E. P., and Krook, M. A Model for Collision Processes in Gases. I. Small Amplitude Processes in Charged and Neutral One-Component Systems. *Physical Review 94* (1954), 511–525.

[17] Bindick, S., and Nachtwey, B. Implementierung eines parallelen Radiosityverfahrens zur Berechnung von strahlungsbeeinflussten Oberflächentemperaturen. *Forum Bauinformatik, Graz* (2007). ISBN:978-3-86068-291-3.

[18] Bitterlich, S., Durner, W., Iden, S. C., and Knabner, P. Inverse estimation of the unsaturated soil hydraulic properties from column outflow experiments using free-form parameterizations. *Vadose Zone Journal 3* (2004), 971–981.

[19] Boghosian, B., Love, P., Coveney, P., Karlin, I., Succi, S., and Yepez, J. Galilean-invariant lattice-Boltzmann models with H theorem. *Physical Review E 68*, 2 (2003), 025103.

[20] Boghosian, B. M., Yepez, J., Coveney, P. V., and Wagner, A. Entropic lattice Boltzmann methods. *Royal Society of London Proceedings Series A 457* (2001), 717–+.

[21] Bouzidi, M., Firdaouss, M., and Lallemand P. Momentum transfer of a lattice-Boltzmann fluid with boundaries. *Physics of Fluids 13* (2001), 3452–3459.

[22] Brezinski, C., and Zaglia, M. R. *Extrapolation Methods. Theory and Practice.* North Holland, Amsterdam, 1991. ISBN 0-444-88814-4.

[23] Buick, J., and Greated, C. A. Gravity in a lattice boltzmann model. *Physical Review E 61* (2000), 5307–5320.

[24] Buick, J. M., Buckley, C. L., Greated, C. A., and Gilbert, J. Lattice Boltzmann BGK simulation of nonlinear sound waves. *Journal of Physics A Mathematical General 33* (2000), 3917–3928.

[25] Buick, J. M., Greated, C. A., and Campbell, D. M. Lattice BGK simulation of sound waves. *Europhysics Letters 43* (1998), 235–240.

[26] (CEI), C. E. I. EnSight - FEA and CFD post-processing visualization. Website, 2008. http://www.ensight.com; accessed on December 30th 2008.

[27] Cercignani, C., Illner, R., and Pulvirenti, M. *The Mathematical Theory of Dilute Gases*, vol. 106 of *Applied Mathematical Sciences.* Springer Verlag, Berlin, Germany; New York, U.S.A., 1994.

[28] Cercignani, C., and Penrose, R. *Ludwig Boltzmann: The Man Who Trusted Atoms.* Oxford University Press, 1998.

[29] Chatzis, I., and Dullien, F. A. Dynamic immiscible displacement mechanisms in pore doublets: theory versus experiment. *Journal of Colloid and Interface Science 91* (1983), 199–222.

[30] Chen, H., and Teixeira, C. H-theorem and origins of instability in thermal lattice Boltzmann models. *Computer Physics Communications 129* (2000), 21–31.

[31] Chen, S., and Doolen, G. Lattice Boltzmann method for fluid flows. *Annual Review of Fluid Mechanics 30* (1998), 329–364.

[32] Chen, Z., Huan, G., and Ma, Y. *Computational Methods for Multiphase Flows in Porous Media.* Society for Industrial and Applied Mathematics, Philadelphia, 2006. ISBN 0-89871-606-3.

[33] Chernyaev, E. V. Marching cubes 33: Construction of topologically correct isosurfaces. Tech. rep., CERN CN 95-17, 1995. Technical Report.

[34] Clift, R., Grace, J. R., and Weber, M. E. *Bubbles, Drops and Particles.* Academic Press, 1978.

[35] Crouse, B. *Lattice-Boltzmann Strömungssimulationen auf Baumdatenstrukturen.* PhD thesis, TU München, 2003.

[36] Crouse, B., Rank, E., Krafczyk, M., and Tölke, J. An LB-Based Approach for Adaptive Flow Simulations. *International Journal of Modern Physics B 17* (2003), 109–112.

[37] Culligan, K., Wildenschild, D., Christensen, B., Gray, W., and Rivers, M. Pore-scale characteristics of multiphase flow in porous media: a comparison of air-water and oil-water experiments. *Advances in Water Resources 29* (2006), 227–238.

[38] da Silva, A. R., and Scavone, G. P. Lattice Boltzmann simulations of the acoustic radiation from waveguides. *Journal of Physics A Mathematical General 40* (2007), 397–408.

[39] Dagum, L., and Menon, R. OpenMP: an industry standard API for shared-memory programming. *IEEE Computational Science and Engineering 5*, 1 (1998), 46–55.

[40] d'Humières, D., Bouzidi, M., and Lallemand, P. Thirteen-velocity three-dimensional lattice boltzmann model. *Physical Review E 63*, 6 (2001), 066702.

[41] Department of Energy (DOE), A. S., and (ASCI), C. I. Visit. Website, 2008. https://wci.llnl.gov/codes/visit/; accessed on December 30th 2008.

[42] d'Humières, D. Generalized lattice-Boltzmann equations. In *Rarefied Gas Dynamics: Theory and Simulations* (Washington DC, 1992), B. D. Shizgal and D. P. Weave, Eds., vol. 159 of *Prog. Astronaut. Aeronaut.*, AIAA, pp. 450–458.

[43] d'Humières, D., Ginzburg, I., Krafczyk, M., Lallemand, P., and Luo, L.-S. Multiple-relaxation-time lattice Boltzmann models in three dimensions. *Philosophical Transactions of the Royal Society A-Mathematical, Physical and Engineering Sciences 360* (2002), 437–451.

[44] Fayon, A., and Happel, J. Effect of a cylindrical boundary on fixed rigid sphere in a moving viscous fluid. *AIChE Journal 6*, 1 (1960), 55–58.

[45] Filippova, O., and Hänel, D. Boundary-Fitting and Local Grid Refinement for Lattice-BGK Models. *International Journal of Modern Physics C 9* (1998), 1271–1279.

[46] Filippova, O., and Hänel, D. A novel lattice BGK approach for low Mach number combustion. *Journal of Computational Physics 158* (2000), 139.

[47] Flekkøy, E. Lattice Bhatnagar-Gross-Krook models for miscible fluids. *Physical Review E 47* (1993), 4247–4257.

[48] Forster, O. *Analysis 3. Integralrechnung im Rn mit Anwendungen.* Vieweg-Verlag, Wiesbaden, 1996. ISBN 3-528-27252-X.

[49] Freudiger, S. *Entwicklung eines parallelen, adaptiven, komponentenbasierten Strömungskerns für hierarchische Gitter auf Basis des Lattice-Boltzmann-Verfahrens.* PhD thesis, TU Braunschweig, 2009.

[50] Freudiger, S., Hegewald, J., and Krafczyk, M. A parallelization concept for a multi-physics lattice boltzmann solver based on hierarchical grids. *Progress in Computational Fluid Dynamics 8*, 1-4 (2008), 168–178.

[51] Frisch, U., d'Humières, D., Hasslacher, B., Lallemand, P., Pomeau, Y., and Rivet, J.-P. Lattice gas hydrodynamics in two and three dimensions. *Complex Systems* (1987), 75–136.

[52] Fuchs, H. V. *Schallabsorber und Schalldämpfer.* Springer Berlin, 2007. ISBN:978-3-540-35493-2.

[53] Geller, S. *Fluid-Struktur-Interaktion mit der Lattice-Boltzmann Methode auf baumhierarchischen Gittern.* Dissertation, TU Braunschweig, 2009.

[54] Geller, S., Krafczyk, M., Tölke, J., Turek, S., and Hron, J. Benchmark computations based on Lattice-Boltzmann, Finite Element and Finite Volume Methods for laminar Flows. *Computers & Fluids 35* (2006), 888–897.

[55] Geller, S., Tölke, J., and Krafczyk, M. Lattice Boltzmann Methods on Quadree-Type Grids for Fluid-Structure Interaction. In *Fluid-Structure Interaction, Modelling, Simulation and Optimisation*, H. Bungartz and M. Schäfer, Eds., vol. 53 of *Lecture Notes in Computational Science and Engineering.* Springer, 2006, pp. 270–293.

[56] Ginzbourg, I., and Adler, P. M. Boundary flow condition analysis for the three-dimensional lattice Boltzmann model. *Journal de Physique II 4* (1994), 191–214.

[57] Ginzburg, I. Equilibrium type and link-type lattice Boltzmann models for generic advection and anisotropic-dispersion equation. *Advances in Water Resources 28*, 11 (2005), 1171–1195.

[58] Ginzburg, I. Variably saturated flow described with the anisotropic Lattice Boltzmann methods. *Computers & Fluids 35*, 8-9 (2006), 831–848.

[59] Ginzburg, I., and d'Humières, D. Local second-order boundary methods for lattice Boltzmann models. *Journal of Statistical Physics 84* (1996), 927–971.

[60] Ginzburg, I., and d'Humières, D. Multi-reflection boundary conditions for lattice Boltzmann models. *Physical Review E 68* (2003), 066614.1–066614.30.

[61] Ginzburg, I., Verhaeghe, F., and d'Humières, D. Two-relaxation-time lattice Boltzmann scheme: About parametrization, velocity, pressure and mixed boundary conditions. *Communications in Computational Physics 3* (2008), 427–478.

[62] Gropp, W., Lusk, E., Ashton, D., Balaji, P., Buntinas, D., Butler, R., Chan, A., Krishna, J., Mercier, G., Ross, R., and Thakur, R. MPICH2 User's Guide, 2006. `http://www-unix.mcs.anl.gov/mpi/mpich`; accessed on December 12th 2008.

[63] Gun, A. Ein Gebietszerleger für parallele Lattice Boltzmann Applikationen auf der Basis von rekursiv-bipartiven Algorithmen für equidistante teilgefüllte 3D-Gitter. *Diploma Thesis* (2002).

[64] Gunstensen, A. K., and Rothmann, D. Lattice Boltzmann model of immiscible fluids. *Physical Review A 43*, 8 (1991), 4320–4327.

[65] Gunstensen, A. K., and Rothmann, D. A Lattice Boltzmann Model for multiphase flow. *Physics of Fluids A5*, 10 (1993), 2557–2561.

[66] Guo, Z., Zheng, C., and Shi, B. Discrete lattice effects on the forcing term in the lattice Boltzmann method. *Physical Review E 65* (2003), 066308.

[67] Haberman, W. L., and Sayre, R. M. Motion of rigid and fluid spheres in stationary and moving liquids inside cylindrical tubes. Tech. rep., US Navy Department, Washington DC, 1958. David Taylor Model Basin Report No. 1143.

[68] Haines, W. B. Studies in the physical properties of soils: V. The hysteresis effect in capillary properties and the modes of moisture distribution associated therewith. *J. Agric. Sci. 20* (1930), 97–116.

[69] Hasimoto, H. On the periodic fundamental solutions of the stokes equations and their application to viscous flow past cubic array of spheres. *Journal of Fluid Mechanics 5* (1959), 317–328.

[70] Haydock, D., and Yeomans, J. M. Lattice Boltzmann simulations of acoustic streaming. *Journal of Physics A Mathematical General 34* (2001), 5201–5213.

[71] He, X., and Luo, L.-S. Theory of the lattice Boltzmann method: from the Boltzmann equation to the lattice Boltzmann equation. *Physical Review E 56* (1997), 6811.

[72] Higuera, F., and Jiménez, J. Boltzmann approach to lattice gas simulations. *Europhysics Letters 9* (1989), 663–668.

[73] Higuera, F. J., Succi, S., and Benzi, R. Lattice Gas Dynamics with Enhanced Collisions. *Europhysics Letters 9*, 4 (1989), 345–349.

[74] Hilpert, M., and Miller, C. T. Pore-morphology-based simulation of drainage in totally wetting porous media. *Advances in Water Resources 24*, 3–4 (2001), 243–255.

[75] Hou, S., Sterling, J., Chen, S., and Doolen, G. D. A Lattice Boltzmann Subgrid Model for High Reynolds Number Flows. *Contributions to Mineralogy and Petrology* (1994), 1004–+.

[76] Inamuro, T., Yoshino, M., and Ogino, F. A non-slip boundary condition for lattice Boltzmann simulations. *Physics of Fluids 7* (1995), 2928.

[77] Izquierdo, S. *Computational Gas Dynamics with the Lattice Boltzmann Method - Preconditioning and Boundary Conditions.* PhD Thesis, Universidad de Zaragoza, 2008.

[78] Janssen, C. Virtual Fluids: a Lattice Boltzmann approach for free surface flow simulations on non-uniform block-structured grids. *in preparation* (2009).

[79] Jensen, H. W. *Realistic Image Synthesis Using Photon Mapping.* AK Peters, 2001. ISBN 1568811470.

[80] Junk, M., Klar, A., and Luo, L. Asymptotic analysis of the lattice Boltzmann equation. *jcp 210* (2005), 676.

[81] Kaestner, A., Lehmann, E., and Stampanoni, M. Imaging and image processing in porous media research. *Advances in Water Resources 31*, 9 (2008), 1174 – 1187.

[82] Karypis, G., and Kumar, V. METIS - A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices. Website, 1998. `http://glaros.dtc.umn.edu/gkhome/views/metis`; accessed on December 30th 2008.

[83] Kehrwald, D. *Numerical analysis of Immiscible Lattice BGK.* PhD thesis, Universität Kaiserslautern, 2003.

[84] Kinsler, L. E., Frey, A. R., Coppens, A. B., and Sanders, J. V. *Fundamentals of Acoustics (4th edition).* John Wiley & Sons Inc, 1999. ISBN 9780471847892.

[85] Kitware, I. Paraview. Website, 2008. `http://www.paraview.org`; accessed on December 30th 2008.

[86] Klinkenberg, L. J. The permeability of Porous media to liquids and gases. *American Petroleum Institute Drilling and Productions Practices* (1941), 200–213.

[87] Kozeny, J. Über kapillare Leitung des Wassers im Boden - Aufstieg, Versickerung und Anwendung auf die Bewässerung. *Sitzungsberichte Akademie der Wissenschaften Wien, Abt. IIa 136* (1927), 271–306.

[88] Krafczyk, M., Lehmann, P., Philippova, O., Hänel, D., and Lantermann, U. Lattice Boltzmann Simulations of complex Multi-Phase Flows. In *Multifield Problems: State of the art* (2000), Springer Verlag, pp. 50–57.

[89] Krafczyk, M., Tölke, J., and Luo, L.-S. Large-Eddy Simulations with a Multiple-Relaxation-Time LBE Model. *APS Meeting Abstracts* (2003), J7+.

[90] Ladd, A. J. C. Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part 2. Numerical results. *Journal of Fluid Mechanics 271* (1994), 311–339.

[91] Lallemand, P., and Luo, L.-S. Theory of the lattice Boltzmann method: Dispersion, dissipation, isotropy, Galilean invariance, and stability. *Physical Review E 61* (2000), 6546–6562.

[92] Lallemand, P., and Luo, L.-S. Lattice Boltzmann method for moving boundaries. *Journal of Computational Physics 184* (2003), 406–421.

[93] Lallemand, P., and Luo, L.-S. Theory of the lattice Boltzmann method: Acoustic and thermal properties in two and three dimensions. *Physical Review E 68*, 3 (2003), 036706–+.

[94] Lehmann, P., Berchtold, M., Ahrenholz, B., Tölke, J., Kaestner, A., Krafczyk, M., Flühler, H., and Künsch, H. Impact of geometrical properties on permeability and fluid phase distribution in porous media. *Advances in Water Resources 31*, 9 (2008), 1188–1204.

[95] Lehmann, P., Krafczyk, M., Gygi, A., Flisch, A., Wyss, P., and Flühler, H. Modelling flow of water and air in reconstructed structures of porous media. In *Proceedings of the 2nd world congress on industrial tomography, Hannover* (2001), VCIPT, University of Leeds, pp. 628–635.

[96] Lehmann, P., Wyss, P., Flisch, A., Lehmann, E., Vontobel, P., Krafczyk, M., Kaestner, A., Beckmann, F., Gygi, A., and Flühler, H. Tomographical imaging and mathematical description of porous media used for the prediction of fluid distribution. *Vadose Zone Journal 5* (2006), 80–97.

[97] Lewiner, T., Lopes, H., Vieira, A. W., and Tavares, G. Efficient implementation of Marching Cubes' cases with topological guarantees. *journal of graphics tools 8*, 2 (2003), 1–15.

[98] Li, H., Pan, C., and Miller, C. Pore-scale investigation of viscous coupling effects for two-phase flow in porous media. *Physical Review E 72* (2005).

[99] Lorensen, W. E., and Cline, H. E. Marching cubes: A high resolution 3d surface construction algorithm. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1987), vol. 21(4), ACM Press, pp. 163–169.

[100] Luo, L.-S. Consistent Initial Conditions for LBE Simulation. *Computers & Fluids 35*, 8-9 (2006), 855–862.

[101] Martys, N., and Chen, H. Simulation of multicomponent fluids in complex three-dimensional geometries by the lattice Boltzmann method. *Physical Review E 53* (1996), 743–750.

[102] McCormick, B. H. Visualization in scientific computing. *SIGBIO Newsl. 10*, 1 (1988), 15–21.

[103] McNamara, G., and Zanetti, G. Use of the Boltzmann equation to simulate lattice-gas automata. *Physical Review Letters 61* (1988), 2332–2335.

[104] Mechel, F. P. *Schallabsorber [sound absorbers], Band I-III (Volume I-III) 1999.* S. Hirzel Verlag, 1999. ISBN:3-7776-0425-9.

[105] Mei, R., Yu, D., Shyy, W., and Luo, L.-S. Force evaluation in the lattice Boltzmann method involving curved geometry. *Physical Review 65*, 4 (2002), 041203–+.

[106] Mei-Juan, Y., Bo-Ming, Y., Bin, Z., and Ming-Tao, H. A geometry model for tortuosity of streamtubes in porous media with spherical particles. *Chinese Physics Letters 22*, 6 (2005), 1464–1467.

[107] Michels, H. Dislin scientific plotting software. Website, 2008. `http://www.dislin.de`; accessed on December 30th 2008.

[108] Miller, C. T., Christakos, G., Imhoff, P. T., McBride, J. F., Pedit, J. A., and Trangenstein, J. A. Multiphase flow and transport modeling in heterogeneous porous media: Challenges and approaches. *Advances in Water Resources 21*, 2 (1998), 77–120.

[109] Moore, G. E. Cramming more components onto integrated circuits. *Proceedings of the IEEE 86*, 1 (1998), 82–85.

[110] Möser, M. *Technische Akustik, 7. Auflage.* Springer Berlin Heidelberg New York, 2002. ISBN:978-3-540-71386-9.

[111] Nguyen, N.-Q., and Ladd, A. J. C. Sedimentation of hard-sphere suspensions at low Reynolds number. *Journal of Fluid Mechanics 525* (2005), 73–104.

[112] Nielson, G. M., and Hamann, B. The asymptotic decider: resolving the ambiguity in marching cubes. In *VIS '91: Proceedings of the 2nd conference on Visualization '91* (Los Alamitos, CA, USA, 1991), IEEE Computer Society Press, pp. 83–91.

[113] Pan, C., Hilpert, M., and Miller, C. T. Lattice-Boltzmann simulation of two-phase flow in porous media. *Water Resources Research 40* (2004).

[114] Polifke, W., and Kopitz, J. *Wärmeübertragung: Grundlagen, analytische und numerische Methoden.* Pearson Studium, 2005. ISBN:3-8273-7104-X.

[115] Priesack, E., and Durner, W. Closed-form expression for the multi-modal unsaturated conductivity function. *Vadose Zone Journal 5*, 1 (2006), 121–124.

[116] Qian, Y. H. Simulating thermohydrodynamics with lattice BGK models. *Journal of Scientific Computing 8* (1993), 231.

[117] Qian, Y. H., d'Humières, D., and Lallemand, P. Lattice BGK models for Navier-Stokes equation. *Europhysics Letters 17*, 6 (1992), 479–484.

[118] Rayleigh, L. J. W. S. Investigation of the character of the equilibrium of an incompressible heavy fluid of variable density. In *Proceedings of the London Mathematical Society* (1883), vol. 14, London Mathematical Society, pp. 170–177.

[119] Roache, P. J. *Computational fluid dynamics.* Hermosa Publishers Albuquerque, N.M., 1972.

[120] Rothman, D., and Zaleski, S. *Lattice-Gas Cellular Automata: Simple Models of Complex Hydrodynamics.* Cambridge University Press, UK, 1997.

[121] Rothmann, D. H., and Keller, J. M. Immiscible cellular automaton fluids. *Journal of Statistical Physics 52*, 8 (1988), 1119–1127.

[122] Sangani, A. S., and Acrivos, A. Slow flow through a periodic array of spheres. *International Journal of Multiphase Flow 8* (1982), 343–360.

[123] Schaap, J., Lehmann, P., Kaestner, A., Vontobel, P., Hassanein, R., Frei, G., de Rooij, G., Lehmann, E., and Flühler, H. Measuring the effect of structural connectivity on the water dynamics in heterogeneous porous media using speedy neutron tomography. *Advances in Water Resources 31*, 9 (2008), 1233–1241.

[124] Schikuta, E., and Message-Passing-Interface-Forum. MPI: A Message-Passing Interface Standard. Tech. rep., University of Tennessee, Knoxville, Tennesee, 1994.

[125] Schiller, L., and Naumann, A. Z. Über die grundlegenden Berechnungen bei der Schwerkraftaufbereitung. *Zeitschrift Des Vereines Deutscher Ingenieure 77*, 12 (1933), 318–320.

[126] Schroeder, W., Martin, K., and Lorensen, B. *The Visualization Toolkit, 4th Edition.* Kitware, Inc., Clifton Park, NY, 2006. ISBN: 978-1930934191.

[127] Schroeder, W. J., Zarge, J. A., and Lorensen, W. E. Decimation of triangle meshes. In *SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1992), ACM, pp. 65–70.

[128] Schubert, H. *Kapillarität in porösen Feststoffsystemen.* Springer-Verlag, ISBN 3-540-11835-7, 1982.

[129] Shan, X. Simulation of Rayleigh-Bernard convection using the lattice Boltzmann method. *Physical Review E 55* (1997), 2780.

[130] Shan, X., and Chen, H. Lattice Boltzmann model for simulating flows with multiple phases and components. *Physical Review E 47* (1993), 1815–1819.

[131] Shan, X., Yuan, X.-F., and Chen, H. Kinetic theory representation of hydrodynamics: a way beyond the Navier-Stokes equation. *J. Fluid Mech. 550* (2006), 413–441.

[132] Silicon Graphics, I. Open inventor. Website, 2006. `http://oss.sgi.com/projects/inventor/`; accessed on December 30th 2008.

[133] Skordos, P. A. Initial and Boundary Conditions for the Lattice Boltzmann Method. *Physical Review E 48*, 6 (1993), 4823–4842.

[134] Skordos, P. A. *Modeling flue organ pipes: subsonic flow, Lattice Boltzmann, and parallel distributed computers.* Phd thesis, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, 1995.

[135] Sterling, J. D., and Chen, S. Stability analysis of lattice Boltzmann methods. *Journal of Computational Physics 123*, 1 (1996), 196–206.

[136] Stiebler, M., Tölke, J., and Krafczyk, M. An Advection-Diffusion Lattice Boltzmann Scheme for Hierarchical Grids. *Computers & Mathematics with Applications 55*, 7 (2007), 1576–1584.

[137] Succi, S. *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond.* Oxford University Press, Oxford, 2001.

[138] Swift, M. R., Orlandini, E., Osborn, W. R., and Yeomans, J. M. Lattice Boltzmann simulation of liquid-gas and binary fluid systems. *Physical Review E 54* (1996), 5041–5052.

[139] Szalmás, L. Slip on curved boundaries in the Lattice Boltzmann Model. *International Journal of Modern Physics C 18*, 1 (2007), 15–24.

[140] Tanikawa, W., and Shimamoto, T. Klinkenberg effect for gas permeability and its comparison to water permeability for porous sedimentary rocks. *Hydrology and Earth System Sciences 3*, 4 (2006), 1315–1338.

[141] Taylor, S. G. I. The instability of liquid surfaces when accelerated in a direction perpendicular to their planes. In *Proceedings of the Royal Society of London A* (1950), vol. 1065, Royal Society of London, pp. 192–196.

[142] Thömmes, G., Becker, J., Junk, M., Vaikuntam, A. K., Kehrwald, D., Klar, A., Steiner, K., and Wiegmann, A. A lattice boltzmann method for immiscible multiphase flow simulations using the level set method. *Journal of Computational Physics 228*, 4 (2009), 1139–1156.

[143] Thürey, N. *Physically based Animation of Free Surface Flows with the Lattice Boltzmann Method*. Phd thesis, Technische Fakultät der Universität Erlangen-Nü, 2007.

[144] Thürey, N. El'beem. Website, 2008. `hhttp://elbeem.sourceforge.net/`; accessed on December 30th 2008.

[145] Tölke, J. A thermal model based on the lattice boltzmann method for low mach number compressible flows. *Journal of Computational and Theoretical Nanoscience 3*, 4 (2006), 579–587.

[146] Tölke, J., Freudiger, S., and Krafczyk, M. An adaptive scheme for LBE multiphase flow simulations on hierarchical grids. *Computers & Fluids 35* (2006), 820–830.

[147] Tölke, J., and Krafczyk, M. Teraflop computing on a desktop pc with gpus for 3d cfd. *International Journal of Computational Fluid Dynamics 22*, 7 (August 2008), 443–456.

[148] Tölke, J. *Gitter-Boltzmann-Verfahren zur Simulation von Zweiphasenströmunge*. PhD thesis, TU München, 2001.

[149] Tölke, J., Ahrenholz, B., Hegewald, J., and Krafczyk, M. Parallel Free-Surface and Multi-Phase Simulations in Complex Geometries using Lattice Boltzmann Methods. In *Third Joint HLRB and KONWIHR Result and Reviewing Workshop*. Fakultät für Informatik, Technische Universität München, 2007.

[150] Tölke, J., Krafczyk, M., Schulz, M., and Rank, E. Lattice Boltzmann Simulations of binary fluid flow through porous media. *Philosophical Transactions of the Royal Society A-Mathematical, Physical and Engineering Sciences 360*, 1792 (2002), 535–545.

[151] Treece, G., Prager, R., and Gee, A. Regularised marching tetrahedra: improved iso-surface extraction. *Computer & Graphics 23*, 4 (1994), 583–598.

[152] Vahala, L., Wah, D., Vahala, G., Carter, J., and Pavlo, P. Thermal Lattice Boltzmann Simulation for Multispecies Fluid Equilibration. *Physical Review E 62* (2000), 507.

[153] van der Sman, R., and Ernst, M. Convection-diffusion lattice Boltzmann schemes for irregular lattices. *Journal of Computational Physics 160* (1999), 766–782.

[154] van Genuchten, M. T. A closed-form equation for predicting the hydraulic conductivity of unsaturated soils. *Soil Science Society of America Journal 44* (1980), 892–898.

[155] van Treeck, C., Rank, E., Krafczyk, M., Tölke, J., and Nachtwey, B. Extension of a hybrid thermal LBE scheme for Large-Eddy simulations of turbulent convective flows. *Computers & Fluids 35*, 8–9 (2006), 863–871.

[156] Visage Imaging, Inc. amira®. Website, 2008. `http://www.amiravis.com`; accessed on December 30th 2008.

[157] Vogel, H.-J., Tölke, J., Schulz, V. P., Krafczyk, M., and Roth, K. Comparison of a Lattice-Boltzmann model, a full-morphology model, and a pore network model for determining capillary pressure-saturation relationships. *Vadose Zone Journal 4*, 2 (2005), 380–388.

[158] Wardlaw, N. C. The effect of geometry, wettability, viscosity and interfacial tension on trapping in single pore-throat pairs. *Journal of Canadian Petroleum Technology 21(3)* (1982).

[159] Washburn, E. W. The Dynamics of Capillary Flow. *Physical Review 17*, 3 (1921), 273–283.

[160] Wellein, G., Zeiser, T., Hager, G., and Donath, S. On the single processor performance of simple lattice Boltzmann kernels. *Computers & Fluids 35*, 8-9 (2006), 910–919.

[161] Wilde, A. *Anwendung des Lattice-Boltzmann-Verfahrens zur Berechnung strömungsakustischer Probleme.* Phd Thesis, Technische Universität Dresden, 2006.

[162] Wildenschild, D., Vaz, C. M. P., Rivers, M. L., Rikard, D., and Christensen, B. S. B. Using x-ray computed tomography in hydrology: systems, resolutions, and limitations. *Journal of Hydrology 267*, 3-4 (2002), 285–297.

[163] Wolf-Gladrow, D. A lattice Boltzmann equation for diffusion. *Journal of Statistical Physics 79*, 5–6 (1995), 1023–1032.

[164] Yang, Z., Dinh, T., Nourgaliev, R., and Sehgal, B. Evaluation of the Darcy's law performance for two–fluid flow hydrodynamics in a particle debris bed using a lattice-Boltzmann model. *Heat And Mass Transfer 36* (2000), 295–304.

[165] Yu, D., Mei, R., and Shyy, W. A multi-block lattice Boltzmann method for viscous fluid flows. *International Journal for Numerical Methods in Fluids 39(2)* (2002), 99–120.

[166] Zapryanov, Z., and Tabakova, S. *Dynamics of Bubbles, Drops and Rigid Particles*, vol. 50 of *Fluid Mechanics and Its Applications.* Springer, 1999. ISBN: 978-0-7923-5347-8.

[167] Zou, Q., and He, X. On pressure and velocity boundary conditions for the lattice Boltzmann BGK model. *Physics of Fluids 9* (1997), 1591–1598.

Index